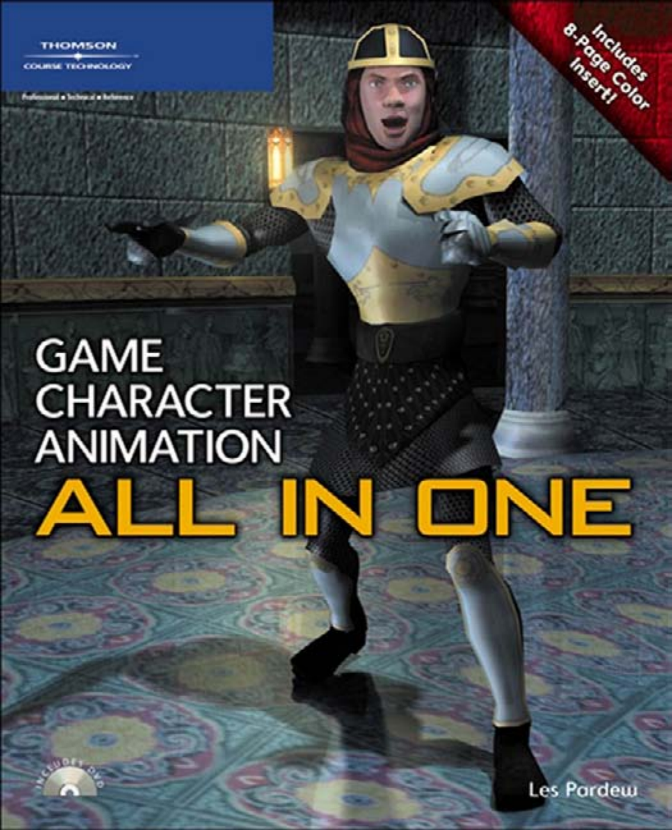
THOMSON
COURSE TECHNOLOGY
Professional ■ Technical ■ Reference

Includes 8-Page Color Insert!



GAME CHARACTER ANIMATION ALL IN ONE

INCLUDES DVD

Les Pardew

Game Character Animation All in One

Les Pardew

THOMSON
COURSE TECHNOLOGY
Professional ■ Technical ■ Reference

ISBN-10: 1-59863-064-4
ISBN-13: 978-1-59863-064-0
eISBN-10: 1-59863-065-2

Library of Congress Catalog Card Number: 2005929822
Printed in the United States of America
07 08 09 10 11 PH 10 9 8 7 6 5 4 3 2 1

THOMSON COURSE TECHNOLOGY™
Professional ■ Technical ■ Reference

Thomson Course Technology PTR,
a division of Thomson Learning Inc.
25 Thomson Place
Boston, MA 02210
http://www.courseptr.com

Publisher and General Manager, Thomson Course Technology PTR:
Stacy L. Hiquet

Associate Director of Marketing:
Sarah O'Donnell

Manager of Editorial Services:
Heather Talbot

Marketing Manager:
Heather Hurley

Senior Acquisitions Editor:
Emi Smith

Marketing Coordinator:
Adena Flitt

Project Editor:
Kate Shoup Welsh

Technical Reviewer:
Dan Whittington

PTR Editorial Services Coordinator:
Erin Johnson

Copy Editor:
Kate Shoup Welsh

Interior Layout Tech:
Interactive Composition Corporation

Cover Designer:
Mike Tanamachi

DVD-ROM Producer:
Brandon Penticuff

Indexer:
Kelly Henthorne

Proofreader:
Steve Honeywell

Acknowledgments

I want to acknowledge and thank all of the many people who have helped me to create this book. My deepest thanks go to my family—especially to my wife, who has put up with living with an artist and all of the ups and downs that that brings. I also want to thank the many mentors who have taught about art. They may never know how much they have influenced my life. Finally, thanks to my editors Kate Welsh, Emi Smith, and Stacy Hiquet, without whose help this book would be impossible.

About the Author

With more than 22 years of experience, **Les Pardew** is a video-game and entertainment-industry veteran. His work in the industry includes more than 120 video-game titles, nine books, and numerous illustrations for magazines, books, and film. He began his career in film animation and later moved to video games, where he has found a permanent home. He currently serves as president of Alpine Studios, which he founded with Ross Wolfley in the fall of 2000.

Les is a prolific artist who loves to work both with digital and traditional media. On the computer, he is an accomplished 3D artist, creating and animating characters for many video games. In traditional media, his first love is drawing, followed closely by oil painting. His favorite subject is people. He can often be seen drawing a portrait or designing a character for a game in his sketchbook.

Les loves to share himself with others by teaching business and art classes at the university level and by authoring several books on art, animation, and game design.

CONTENTS

Introduction

For the artist, designing characters is probably one of the most exciting parts of creating video games. If you are new to this aspect of video-game development, welcome. I hope you have a wonderful time using your imagination to create many fantastic video-game characters and creatures. If you have already explored video game–character development, then I hope this book will give you added information in that area.

As you can see from the title of this book, it is devoted to covering the development and animation of video-game characters in detail. There is enough basic-level material in the book for even the raw beginner to get a good foundation in character development. Also included are some more advanced character-creation and animation techniques. Hopefully, this book will be a good all-around resource for you and your artistic development.

Character development is fascinating for many reasons—especially in video games. That's because in a game, the characters become almost living entities. With almost any other medium, the character only follows a script. In video games, however, characters can use logic and complex artificial intelligence to act and react to situational events. This added dimension gives the video-game designer the ultimate in creative expression.

From sketch to completion, the video game–character designer has a huge challenge: to create exciting and meaningful characters that can react naturally in a virtual world. Advances in video-game technology are also opening up even more opportunities to create more-detailed characters and add greater depth to their personalities.

As you read this book, you will see that even though it covers many aspects of developing and animating a character, what's discussed here is only the beginning. There is still a vast amount of knowledge to be gained in this field because character development is a very big subject. In a way, it is similar to creating an actual person or creature. Anyone who has had children knows how complex that can be.

As you read and work through the topics and examples in the book, I recommend that you try to follow the step-by-step instructions so that you get the most out of it. Learning about art is really a process of doing; following the examples will help to make the concepts more concrete in your mind. It will also give you a great chance to learn some important software programs that you can use in your professional career.

Included with this book is a DVD containing a number of professional software programs. They are either trial versions or learning versions of the software, and are great for learning how to use them. I cover many of the programs in the book, but some of them are extras that you can use as a bonus. Also included on the DVD are all the pictures from the book. If you have any trouble seeing the demonstrations in the book or would like to see them in full color, you can refer to the pictures on the DVD.

I hope this book is just the beginning to your development as a great character development artist. Good luck in your art.

This page intentionally left blank

CHAPTER 1

Game Characters

Creating 3D characters for video games is fun and exciting. This book is your gateway to the fascinating world of game-character development. It covers the topic of character development from the very first basic idea all the way through the finished character with animation ready to import into a game. This first chapter is an introduction to the world of game-character creation. It covers the following topics:

- What is a game character?
- How are game characters categorized in games?
- How are game characters viewed?
- Where did game characters come from?
- What can be expected in the future for game characters?
- What tools are used to create characters?
- How are 3D graphics used to create games?

What Are Game Characters?

With the possible exception of some puzzle games, almost every game has characters. A game character is an intelligent person, creature, or machine in a game. Actually, a game character can be almost anything—as long as it has some

Figure 1.1
This scene shows two game characters.

intelligent function that acts with or reacts to the player. Some game characters are beautiful, and some are ugly. Some are stylized cartoon characters, while others, like the ones in Figure 1.1, are very realistic. There is almost as much variety in game characters as imagination will permit.

Game characters include all intelligent people or creatures in a game. *Intelligent* means that the character is either controlled by the player or by the game software. Characters that are controlled by a player derive their intelligence from the player. Characters controlled by software derive their intelligence from the software. In some games, the characters use a combination of software and player control.

Player-controlled characters can be in single-player games and in multi-player games. In a single-player game, the player controls the main character directly. In multi-player games, there are several players, each of whom controls a different character. Often, games give players the ability to customize their characters. Some games go so far as to include a character editor or creator that allows the player to alter his or her character in great detail.

The game software controls characters that are not controlled by a player. The specific software that controls characters in games is called *artificial intelligence,* or

AI for short. AI in games is becoming increasingly complex, to the point that the player can interact with AI-controlled characters as if those characters were real.

Some games have characters that are controlled by a combination of player input and game software. This is often the case for real-time strategy games in which the player may control hundreds of characters at the same time. In these types of games, the player may control the characters' location and even be able to give commands for certain tasks. The game software then has the characters perform the task as long as they are not interrupted. If interrupted, the characters will react automatically to the interruption without the player needing to do anything.

Characters in games are becoming more and more lifelike. They don't just run around and shoot things anymore; they can also have emotion and sophisticated reactions to the player. An angry character will not just attack with sword drawn, but will also have facial expressions and body language to emphasize the attack. He may yell at the player or even call other game characters to help him fight. Characters in games can have a full range of emotions. This increase in sophistication has increased the demands on designers and game artists to develop interesting and engaging characters.

In the past, all the game artist had to do was create a character and give it a few simple animations, like run, shoot, and die. Those simple games are rapidly becoming a thing of the past. Today, the game artist not only has to design the character but also has to animate the character with a wide range of actions and expressions. Say, for instance, that a game is a military simulation. One of the team members may have a hidden fear of explosions and freak out when the team starts to get shelled. The game artist is challenged with creating animations to depict that character's erratic behavior in addition to the normal game animations.

These increased demands on the game artist have created a new opportunity for artists entering the field who have the ability to create sophisticated models and complicated animations.

Types of Game Characters

Games have many different types of characters. The following is a list of categories into which game characters can be broken:

- Player characters
- General non-player characters
- Enemies

Each type of character can be further broken down into sub-types. For example, player characters can be characters controlled by a single player or, in the case of a multi-player game, characters controlled by other players. Non-player characters, or *NPCs* for short, can be allies, facilitators, or decoration. Enemies can be rivals, aggressive, passive, or traitorous. This breakdown is only one of many ways to categorize characters within a game.

Player Characters

A *player character* is a character that is controlled by a player. In many games, there is only one player character—the one being controlled by you, the player. Some games, such as real-time strategy or team-sports games, have the player control multiple characters at once with the assistance of sophisticated AI. In these team games, the player may control individual characters directly, but most of the time, the player characters are doing their jobs as assigned by the player.

The amount of control the player has over a character will depend on the type of game being played. For example, a fighting game will give the player a lot of control over the character. The player will be able to control the movement and actions of the character in detail. Adventure games are similar in that the player will be able to control the character's actions to navigate through the world. Other games, like real-time strategy games, give the player control over the character's position and over assigning jobs. The player's characters then intelligently act upon the environment and upon game events.

Multi-player games have several player-controlled characters in a single game. Some of these games are made up entirely of player-controlled characters. A multi-player game may have hundreds of player-controlled characters, all interacting in a single game environment.

Many multi-player and single-player games allow the player to customize his or her character, some going so far as to include a character editor with the game. Character editors are challenging for the game artist because instead of creating a single character with a given appearance, the game artist has to anticipate multiple variations in costume, hair, and accessories. For example, if the game will have an option to change the character's hair, the artist must model several hairstyles for the character, each one interchangeable. Instead of modeling the character's hair one time, the artist may have to model several hairstyles and save them as separate objects so the programmers can swap them out as directed by the player.

Figure 1.2
Player characters are usually more important than other characters.

Player characters usually require more detail than other characters because they are at the center of the player's interest and will be scrutinized more extensively than other game characters. As a result, many times the player's characters will have a higher resolution in detail and construction, as shown in Figure 1.2.

Non-Player Characters

Technically, any character that is not a player-controlled character is a non-player character. That said, it is useful to divide NPCs into two groups: general NPCs and enemies. Enemies are distinct in that players have to react very differently to them than they do to other characters in the game. I will talk about enemies in a moment.

General NPCs are usually not overtly hostile to the player unless provoked, although they may be indifferent or have very limited reaction to the player. And of course, if a general NPC *is* provoked, it may become an enemy rather than a general NPC. Likewise, if the player pacifies an enemy, that enemy could change roles and become a general NPC.

Some games are filled with general NPCs. Team sports games, for example, have NPCs that play the role of teammates, cheerleaders, spectators, announcers, and so forth. Members of the opposing team, however, are better classified as enemies because although they might not necessarily try to kill the player character, their goal is to defeat the player.

Probably the best examples of games with general NPCs are adventure games. In an adventure game, the player interacts with several general NPCs on many different levels. Some of the characters in the game may be helpful to the player by giving information or items to the player. Others, such as store clerks or other types of merchants, may be indifferent yet help in their own way. Some characters may be nothing but decorative, giving no useful information and acting only as a distraction.

In some games where the general NPCs change roles depending on the actions of the player, the identity of the character as general NPC or enemy may not be known. For example, an indifferent merchant shown in Figure 1.3 may change to

Figure 1.3
The friendly green guy just wants to help.

and become an ally if the player completes a quest or helps solve a problem important to that merchant. On the other hand, the merchant may become an enemy if the player tries to steal something from his or her shop.

Enemies

Enemies are all non–player-controlled characters that try to keep the player from winning the game. The members of the opposing team in a sports game are enemies. The evil creatures in a horror game are enemies. The other drivers in a racing game are enemies. The vicious alien trying to kill the player in a first-person shooter is an enemy. Given this definition, other player-controlled characters in a multi-player game may also be considered enemies.

In some games, enemies are very intelligent and cunning, while in others they may just be aggressive. A lot depends on the type of game and the type of enemy. For example, foot soldiers in a real-time strategy game may not have as much intelligence as the hard-to-build mega-tank unit. Differences in intelligence can offer a variety of challenges for the player.

Sophistication in intelligence can take on many forms in different games. In a football game, the opposing team may be very good at play calling and disguising play coverage. In a shooter, the enemies may be very good at ganging up on the player. In a fighting game, the opponent may have several combo moves. Games need to provide a challenge for the player or they become dull; the enemies are usually the ones who supply this challenge.

In designing enemies, the look of the character is often as important as what the character does in the game. In some games, the characters should be intimidating to the player, causing him or her to feel a sense of accomplishment when that enemy is defeated. Sometimes the intimidation is from an imposing physical appearance, like in Figure 1.4. Other times the intimidation is subtler, like in a quiz game where the opponent needs to appear intelligent. Other games may want the player to think an enemy is an ally, so the enemy will appear friendly at first and then betray the player later after earning the player's trust.

Designing Characters

The game artist needs to have a very good understanding of the characters in a game to really do a good job designing them. The game artist needs to understand the character's role and create a character that fits that role. A sleazy police

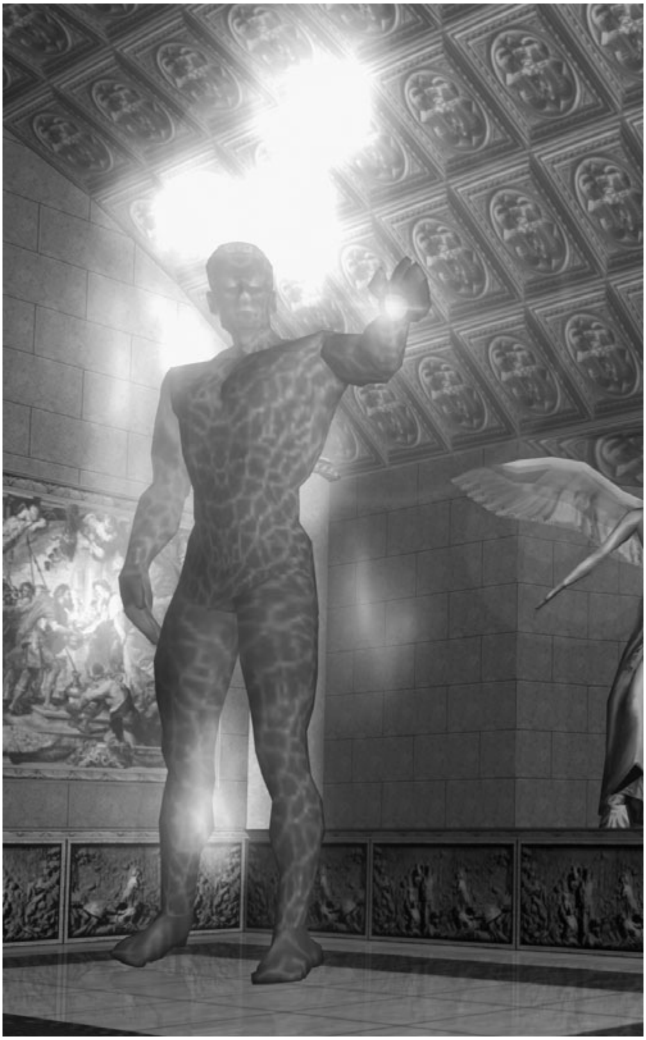


Figure 1.4
I think he has fire in his eyes.

informant should not wear a three-piece tailored suit. A military guard is not very menacing if he is slight of build. An obese wide receiver would look kind of weird. Characters should fit the role for which they are intended. This is not to say the game artist does not have any creative freedom, but rather to say that the character needs to be well matched to its purpose like the pirate girl shown in Figure 1.5. In Chapter 2, "Developing Character Ideas," I will cover character design in detail.

Figure 1.5
Good characters start with a good design.

Viewing Characters in a Game

The best way to begin any discussion on game art is to clarify how art is displayed in a game. Most people play games on a computer, handheld device, or console game system. The pictures you see in games on these systems are made up of small, colored, square dots of light called *pixels*. More precisely, a pixel can be defined as the smallest controllable segment of a display. Back when computer games first came out, the size of video-game pixels was very large, and they appeared as big blocks of color. As technology has advanced, the size of pixels has shrunk to the point that in some game systems, it is difficult to see a single pixel.

Pixels are small dots of colored light that make up pictures on a computer screen. I am repeating this because it is very important. In traditional art, artists work mostly with the reflected light of a painted surface. For games, artists work with pure light as it is displayed on a screen as opposed to painting on a canvas. This fundamental difference takes a little getting used to, particularly in the area of color (see Figure 1.6).

A game artist uses colored light to create images. Most other forms of art are reflected light. For example, when a person looks at an oil painting, he or she sees colors that are reflected from light in the room. On the other hand, when a person looks at the same painting displayed on a color monitor or TV he is looking at direct light, not reflected light.

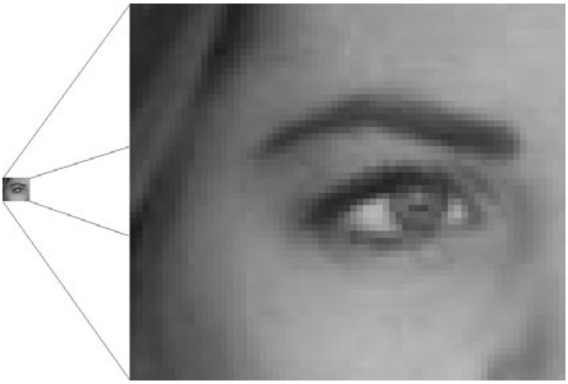


Figure 1.6
This image is enlarged to show the individual pixels.

Reflected light is not as bright and vibrant as direct light; however, we live in a world of reflected light. When you are creating game art, it is important to remember that the art will look unrealistic or cartoon-like if you don't take care to reduce the intensity of the color to match how the object looks in real life.

A Historical Overview of Computer Art and Game Characters

The first computer art goes back to even before the pixel was developed. Back when computers were in their infancy, the pioneers in the industry used printouts of a device called an oscilloscope rather than computer monitors. Some of the earliest computer games were TeleType games in which the players sent updates to each other as printouts. The characters in these games were mostly described using text, but occasionally they were drawn, as in Figure 1.7.

As display technology advanced, digital graphics creation moved from the printer to the computer monitor. Early monitors were monochromatic and designed for text, not for graphics. Regardless, artists found ways to create art even if it was only in one color.

Most people mark 1963 as the beginning of computer art, when Ivan Sutherland invented Sketchpad. Sketchpad allowed artists to use a light-pen device to work on vector images on a vector graphics display monitor. *Vector graphics* are lines and curves derived from mathematical formulas like those pictured in Figure 1.8.

Later in that same decade, the raster image was invented, and pixels came into being. A *raster* is the pattern of dots that form an image. Raster images are made

```
  XXXX
  XXXXX
  XXXXX
   XXX
 XXXXXXX
XXXXXXXXX    XXXX
XXXXXXXXXXXXXXX
XXXXXXXXXXXXXX
XXXXXXXXX
XXXXXXXXX
XXXXXXXXXX
XXXX  XXXX
XXXX   XXXX
XXXX   XXXX
XXXX   XXXX
XXXXXX XXXXXX
XXXXXX XXXXXX
```

Figure 1.7
An early TeleType character.

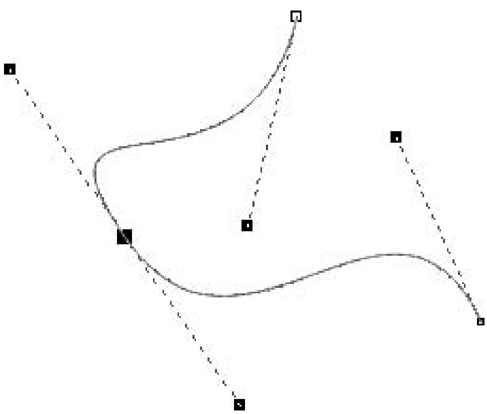


Figure 1.8
Mathematical curves are used to form vector graphics.

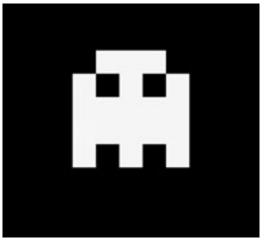


Figure 1.9
Early game characters were one color and very blocky.

up of horizontal lines of pixels, and the number of horizontal lines and the number of pixels in each line determine the resolution of the computer monitor.

The first raster images were very blocky because those early computers were not powerful enough to project very many pixels. You can see clearly the progress of raster images in the evolution of characters in games. Figure 1.9 shows a very early game character. The image is in one color and the pixels are very large. This character is only five pixels wide and five pixels high.

The Move to Color

It didn't take long for scientists to learn how to change the colors on computer monitors. Computers were still considered business or research tools, not art tools, but the artist was not to be denied. Programmers with interest in creating graphics on computers started writing programs to create computer art.

During this time, another important event took place: The personal computer was introduced to the market. These early machines were not very powerful compared to the big mainframe systems, but they did do something that changed the face of computing forever. They put computers in the hands of millions of people, some of whom were interested in creating art. Many of these people were the ones who started writing graphics programs for the personal computer, which were the forerunners of the programs we use today. Console game systems that used the family TV as a monitor were also introduced and companies like Atari and Intelevision became the pioneers of the console game system.

Figure 1.10 shows another early game character. Notice that he has more pixels and different colors than the character in Figure 1.9. He is 16 pixels high and seven pixels wide.

This character has four colors. In those days, many systems were limited in how many colors they supported. Some had set colors, such as the old IBM PCs with

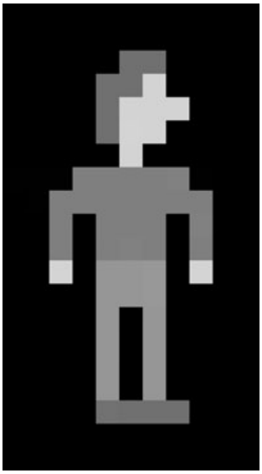


Figure 1.10
Early color game characters were limited in their number of colors.

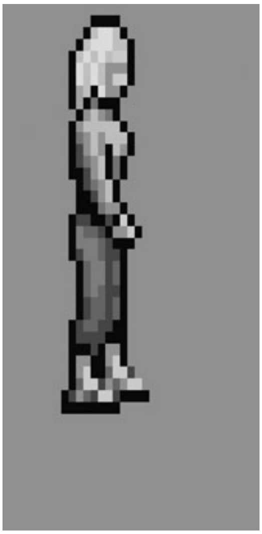


Figure 1.11
When variable colors became available, game characters started taking on a sense of volume.

their CGA and EGA color palettes. Others like Apple, Commodore, and Atari, allowed some variations in color.

Progress was being made every day in computer graphics. With each succeeding year, systems were introduced that boasted higher resolutions and more colors. Figure 1.11 shows a game character with a 16-color variable set of colors, or

Figure 1.12
Greater resolution meant more detail for computer-game characters.

palette. More colors meant artists could start shading their characters. Notice that the game character is starting to take on a sense of volume.

In the late ’80s, games were becoming more realistic, but they still looked blocky. The same thing was true for all computer art. Variable colors gave computer artists a chance to start creating more realistic digital pictures. During this time, desktop publishing was developed, the laser printer was invented, and people started to look at the computer as an art tool rather than just as a business machine.

With a greater emphasis on graphics, personal computer systems entered a whole new path of evolution, bringing about better display systems and higher resolutions. The game characters went from being about 30 to 60 pixels high to around 200 pixels high, and displays went from 320×200 to 640×480 resolutions. Figure 1.12 shows a game character at about 180 pixels high using an 8-bit, or 256-color palette.

More Power Brings Better Tools

Once computers became popular for desktop publishing, the race to create better graphics tools started in earnest. Now developers had a real market for their programs because artists were buying computers and software in great numbers. This paved the way for companies such as Adobe and Corel to create software for personal PCs and sell the software directly to the general public rather than only to art-production companies.

Computer manufacturers and chip makers were intrigued by this new set of computer users. Artists wanted the fastest systems with the best graphics capabilities. They were driving the high end of computer usage and represented a substantial market for the most expensive machines.

Within a few short years, the entire landscape of personal computers changed. System memory and processor speeds increased dramatically. Digital input devices, such as tablets, were developed. Computer monitors became larger. Graphics cards were developed, which gave computer systems added graphic punch. These rapid advances moved the personal computer from a tool for hobbyists to a necessity for graphics professionals.

3D Makes Its Move

Three-dimensional computer art started in applications such as military simulators. These simulators were custom-built computers tied to hydraulic systems and used to simulate aircraft or vehicle motion. The first simulators were not photo-realistic, but they did give the trainee the feeling of being in a real environment. The advantage of these simulators was that the trainee could make mistakes without the danger of getting hurt or destroying valuable military vehicles or airplanes.

It didn't take long for game designers to start experimenting with 3D graphics. Because the initial rendering power of home computers was limited, the characters for these games were created with very few polygons, like the one shown in Figure 1.13. Although they were very blocky at first, computer games with 3D characters were readily accepted by the public. Game players liked the idea of games with 3D characters and environments.

As graphic power increased, it soon became apparent that computer art could be used in both motion-picture and video-game productions. At first, digital 3D art was used to augment physical models or to enhance special film effects. As time went on, however, digital 3D art started to be used more and more, until entire

Figure 1.13
The first 3D game characters were very blocky.

movies, such as *Toy Story* and *Dinosaur*, were done with computer art and no live actors.

Most of the initial 3D graphics for motion pictures were created on high-powered systems called *workstations*. These graphic workhorses were specially designed computer systems with the most powerful graphics hardware available. Workstations are specialized computers with specialized hardware and software dedicated to a specific purpose like desktop publishing or 3D modeling. They were significantly more expensive than personal computers, and they ran specialized software that was also very costly. The era of the workstation, however, was destined to be short-lived.

As more and more people purchased personal computers, the prices for computers started to fall. This was driven in large part by competition from computer and chip manufacturers to get computers into as many homes as possible and the fact that mass production made individual systems less expensive to build. So while personal computers were advancing daily in power and sophistication, they were also becoming more affordable. The more expensive process of building workstations had no chance of competing with the onrush of the PC.

As PCs became more powerful, high-end 3D software migrated from the workstation to the standard off-the-shelf PC. This move opened the door for computer artists everywhere to create beautiful 3D computer art.

Digital 3D art was also entering the computer- and video-game world. Home PC and video-game consoles were rapidly gaining in power. In the mid-1990s, the video- and computer-game industry moved from primarily using 2D art to using 3D art for most games, and names like "Playstation" and "N64" became household words. Figure 1.14 shows a character from these systems.

Figure 1.14
Game characters in 3D became very common in the mid-1990s.

PCs were not the only systems to advance rapidly in computing power. Video-game systems were also becoming more powerful. By the late 1990s and early 2000s, game systems were capable of using almost lifelike graphics. Figure 1.15 shows a character from one of these systems.

Figure 1.16 displays a timeline of the evolution of the game character to show how computer art and game characters have advanced over the years.

Figure 1.15
3D characters replaced 2D characters in video games.

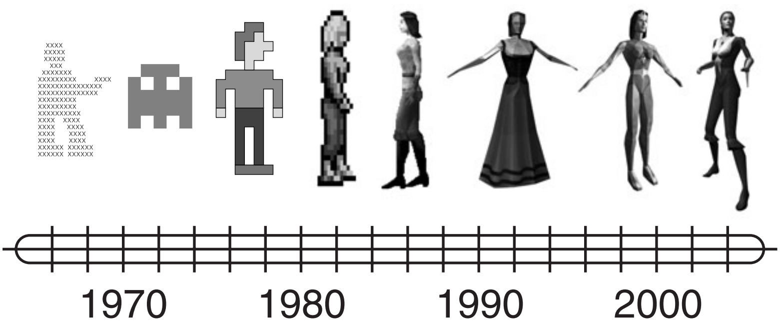


Figure 1.16
This timeline shows the advance of digital graphics in video games.

Looking to the Future

New machines like the Xbox 360, Revolution, and Playstation 3 show great promise to vastly increase the quality of character art in gaming. These new systems allow the game artist to approach lifelike realism in character creation. Things like dynamic hair, mapping effects, and detailed geometry that were impossible on the older systems are becoming a reality. Game artists who want to be on the forefront of these advances will have to learn ever-more advanced modeling and animation techniques. Fortunately, the software programs for creating this type of advanced art are already in place.

Moving forward, there are limitless possibilities for creating computer art. Many off-the-shelf PCs are powerful enough to handle complex digital composition. Software—such as Adobe Photoshop, Adobe Illustrator, Corel Painter, CorelDRAW, Autodesk Maya, Autodesk 3ds Max, and many others—has come down in price, as has hardware, effectively creating an environment that favors the artist. Artists are free to create incredible art with advanced tools, and the future looks like it will only get better for the computer artist.

2D and 3D Art

Artists use a variety of computer programs to create art for games. These programs fit into two basic categories—two-dimensional (2D) programs and three-dimensional (3D) programs. Two-dimensional programs are the easiest to understand because computer screens and video game screens are basically flat. A 2D art program directly manipulates pixels on screen. Many of these programs are very sophisticated, and some even simulate natural media, such as airbrushes, oil paints, or even watercolors.

Three-dimensional programs create virtual 3D objects used in the creation of 3D game characters and game worlds. In this book, I will be using Autodesk's Maya to create 3D characters. Maya is a great program for creating 3D art for games. It has a huge following with developers in the industry—to the point that almost every game artist who deals with 3D game art has to have at least a working knowledge of the program.

Because this book focuses on creating game-character art using Maya, most of the information will relate to 3D art in Maya. Although Maya is only one program, it is similar to other 3D programs, and learning how to use it will help with understanding 3D and 3D art programs in general. Maya also has a personal learning edition, which is given away free, so you will be able to follow along with the examples in the book.

A large part of 3D art for games is created in 2D art programs. Included with this book are trial versions of some of these programs on the CD to help you get started. In addition, one 3D program is included. Here's a run-down of what's on the CD:

- Corel Painter
- CorelDRAW suite
- Jasc Paint Shop Pro
- Maya PLE

Hint

Take some time to explore and become familiar with the art programs on the accompanying CD-ROM. Each program is a professional tool. The better you understand these programs, the more you will gain from the projects in this book.

In later chapters, there will be several specific exercises that deal directly with these programs. They are all programs that are used regularly in my own work, and each one is a true professional program.

Using Painting and Drawing Programs

Drawing programs of today hearken back to some of the first art programs ever created for computers. These programs are based on vector graphics and actually keep track of each line that is drawn. The CD contains CorelDRAW, which is a very powerful and complete vector-drawing program. Other good vector programs include Adobe Illustrator and Macromedia FreeHand.

Paint programs differ from drawing programs in that they are based on raster graphics rather than vector graphics. Some of the most popular painting programs are those used in photo manipulation. Adobe Photoshop is likely the most widely used paint program in the game industry. It is very powerful and has extensive tools and functions for working with photographs and for art creation in general. The CD that comes with this book contains two programs that are comparable to Photoshop: Corel PHOTO-PAINT, which is part of the CorelDRAW suite, and Jasc Paint Shop Pro.

Figure 1.17
Use a tablet for digital drawing. (Photo courtesy of Wacom Technology. All rights reserved.)

Game artists use painting programs to create 2D art for games. Two-dimensional art is often created by the artist from scratch instead of by manipulating other art or photographs. For this reason, the CD that comes with this book contains a painting program—Corel Painter. This program is great for creating art. It has some very powerful features that allow you to use tools that simulate natural drawing and painting techniques.

Drawing on a computer is much like drawing on paper—*if* you have the right hardware. I like to use a tablet, like the Wacom Intuos tablet shown in Figure 1.17. Tablets are nice because they can be held in any position, like a sketchbook. The tablet uses a stylus pen that weighs about the same as a pencil, and feels a lot like one too. When you move the stylus over the tablet, the cursor on the computer screen moves along with it. The stylus has a pressure-sensitive tip, which paint programs use to simulate the pressure the artist uses in drawing. You can also use the stylus to execute commands on the screen; simply touch the

Figure 1.18
New technology allows artists to draw directly on the screen. (Photo courtesy of Wacom Technology. All rights reserved.)

tip of the stylus on the tablet to select an option onscreen. New touch-screen technology is also available for those who want to work directly on the screen; the Wacom Cintiq tablet/monitor shown in Figure 1.18 is a good example.

Drawing with a mouse is clumsy at best. If you are serious about creating art for games, I highly recommend getting a digitizing tablet because it helps make the drawing and painting process on the computer more natural. Don't worry if you don't have one, though. You can still complete the projects in this book because all of the art programs used in the projects work with a mouse.

3D Art in Games

Most video and computer games today use 3D art to create the game experience. To really understand how to create 3D art, you first must understand how it is used in a game. First, the image you are viewing on a TV or computer monitor is not really 3D because, as I mentioned earlier, the display is a flat surface made up of pixels. Instead, the images you are seeing are really *pictures* of a 3D game world, viewed in rapid succession. In other words, the game system is processing a virtual 3D environment and creating images of that environment, which it sends to the display for viewing. It is actually a very complex process that takes a lot of computer processing power. Let's break it down a little so it is easier to understand.

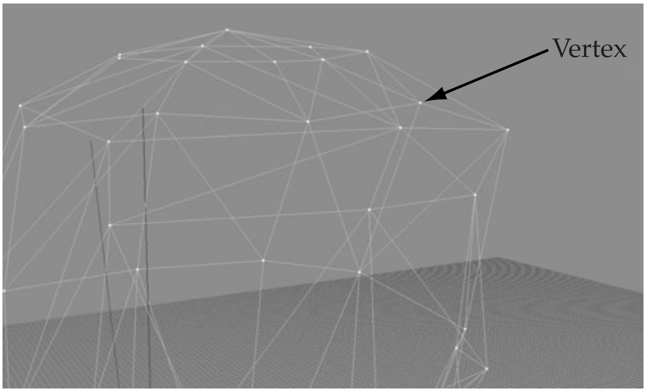


Figure 1.19
A vertex is a single point in virtual space.

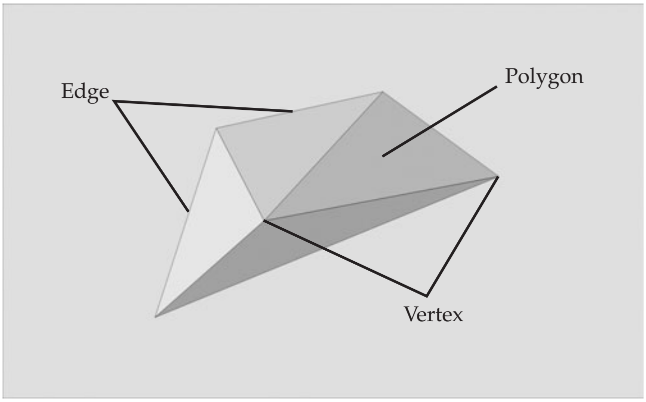


Figure 1.20
Polygons are flat planes in virtual 3D space.

The Vertex

The basic building unit of a 3D game is the vertex. A *vertex* is a single point in virtual 3D space. Figure 1.19 shows a vertex as a yellow dot. It has no dimension in virtual space; it is just a single point. It is in reality a location. To even begin to calculate a 3D object, the computer has to start with a location. From that location, the computer then can define other locations, or vertices, that define a 3D object.

The Polygon

Polygons are flat planes in virtual 3D space. Vertices make up the corners of a polygon plane. The lines that connect the vertices to make up the polygon plane are called *edges*. Figure 1.20 shows several polygons, vertices, and edges.

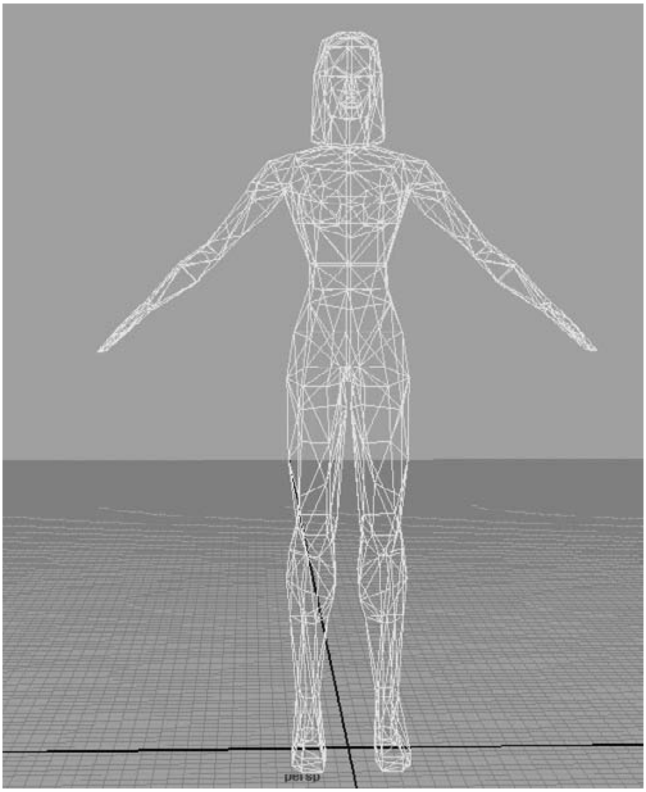


Figure 1.21
Wire-frame models have no surfaces.

Because a polygon is a flat plane, it has no real thickness. It is simply a plane that indicates the joining of three, four, or more vertices. It is important to understand that polygons don't have any depth because polygon models are almost always hollow wire frames. *Wire frame* is the term used to describe a 3D model without surfaces. The term can also refer to a mode of viewing the model without surfaces. The wire frame consists of vertices connected by edges forming polygons. Figure 1.21 shows the wire frame of a character model.

By arranging the polygons so they all connect with each other, the game artist can create what appears to be a 3D object. Figure 1.22 shows a simple human head created in polygons.

Note

There are solid-modeling programs used in professions such as industrial design, but the calculations for solid models are more extensive and time consuming than those for wire-frame models.

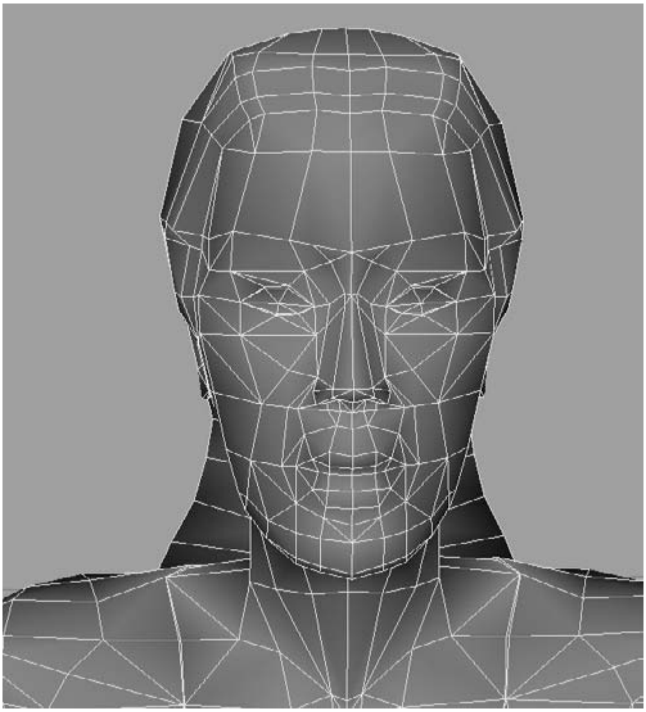


Figure 1.22
Polygon models can appear to be solid.

Textures

Textures are 2D images that are applied to a three-dimensional model to give the model surface detail. In real life, every surface has a texture. Sometimes the textures are only a color, while other times they might be very complex, such as the bark of a tree. Take a quick look around and study some of the many textures you see in everyday life. Figure 1.23 shows polygon head with a texture added.

Textures are often used in games to help reduce the number of polygons needed to create models. This is very important, because the number of polygons used in creating a character affect the speed at which the character can be rendered by the game engine. Figure 1.24 shows a wooden door. You could create this door using nothing more than color and geometry, but that would be an extreme waste of

Figure 1.23
A texture is added to the head model.

Figure 1.24
Adding a texture for the door can save processor time.

computer processor time that you could devote instead to more critical game needs. It would also be an extreme waste of *your* time. A better method would be to create the appearance of the wooden door in a 2D picture and paste that picture on a relatively simple object.

You will notice on close examination that every surface in nature has some qualities that you can fit into a few specific categories:

- **Color.** The actual color of a surface
- **Roughness.** The rough or smooth quality of a surface
- **Translucency.** The amount of light that passes through a surface
- **Reflectivity.** The amount of light that bounces off a surface
- **Luminance.** The amount of light generated by a surface like a florescent tube

Each one of these qualities or attributes is part of what gives the surface the look and feel it has. To make a 3D model look believable in a game, the artist needs to capture the inherent qualities of the surfaces he is trying to depict by creating textures that match those surfaces as closely as possible. The metallic sheen of a kitchen appliance has a very different look from a weathered fencepost. The hard gray of a sidewalk is very different from the spiky look of the lawn right next to it. Much of the quality of a surface in a game is created by the 2D art used to create the texture, but some is created within the 3D program.

Programs like Maya add a variety of attributes to surfaces that range from transparency to glossiness. Maya creates a series of rendering nodes used to add qualities to a surface. For example, transparency can be added to a surface, giving it a smoked-glass quality. A bump map can be added to a weathered metal material, simulating peeling paint. Specularity can give a surface the feeling of shiny metal. Figure 1.25 shows a marble ball sitting on a wooden table; notice that the marble is reflective and the wood has a glossy finish.

Figure 1.26 adds transparency to the ball.

Not only can surface attributes be added, but other effects can also change the very nature of an object. Figure 1.27 shows what happens when an internal glow is added to an object with transparency, giving it the look of a light.

Only recently has rendering power increased enough in game hardware that surface qualities are playing a role with objects and characters in games. Historically, game models dealt mostly with color and ignored the other aspects of surface characteristics because the systems were too slow to handle the processing requirements of things like reflection and transparency. Although these advances are still new, they will play an increasing role in character design in the future.

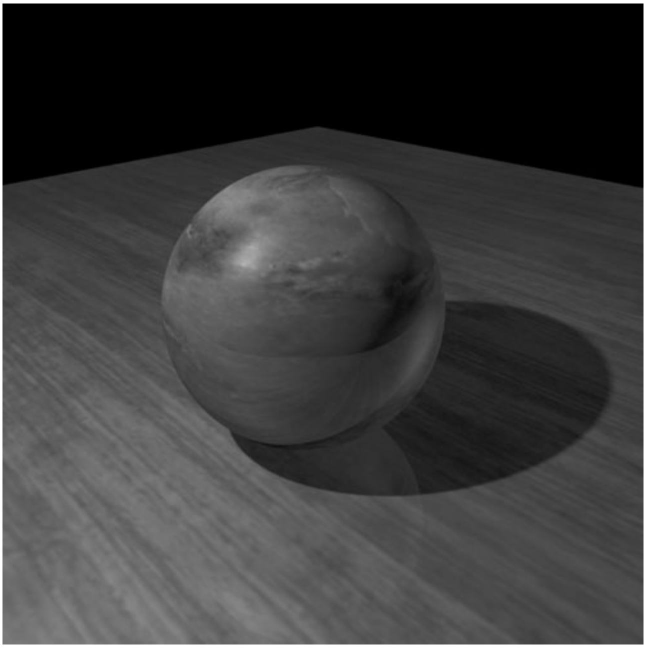


Figure 1.25
The marble ball and table have reflective surfaces.

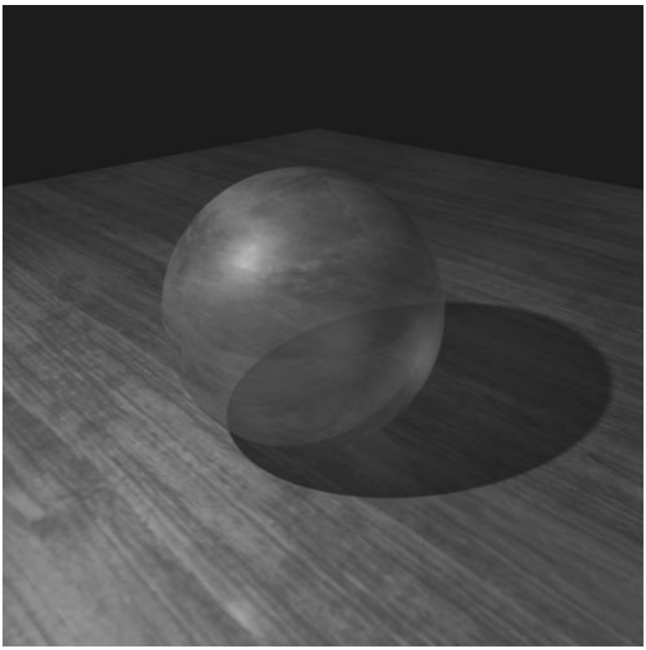


Figure 1.26
Transparency is added.

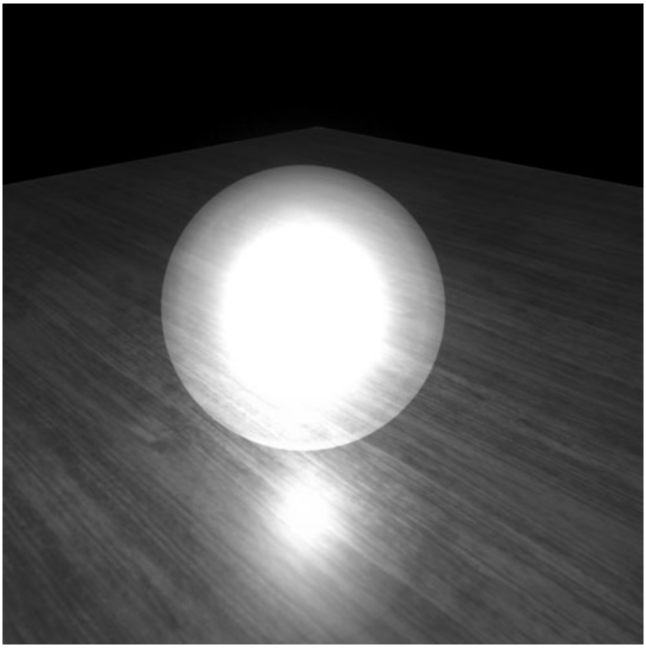


Figure 1.27
Now the ball is a light.

Lighting and Reflections

An important part of developing realistic game environments is developing qualities and attributes you see in normal life. How an environment is lit will play a big role in how realistic that environment will look. Another important factor of an environment is its reflective nature. A shiny object should reflect what is around it.

Lighting and reflections take time and power to render. The more lights a scene has, the more processing time it takes to render that scene. Reflections are particularly rendering-intensive because they require a multitude of complex calculations. As a character artist, it is very important that you understand the limitations of your intended game system and keep your character designs within those limitations.

Lighting

Although the environment artists primarily use lighting in a game, it is important for the character artist to understand how a scene might be lit. Lighting is used in a game to illuminate the game geometry; it is updated every frame so it simulates the real world. When designing the lighting for a game, you need to understand two important things: where and what. By where, I mean the location of the light. By what, I mean the type of light. Some of the more common types of light are as follows:

- Point light
- Directional light
- Ambient light
- Colored light

Point Light A *point light* is similar to the light given off by a light bulb. The light projects in all directions, but is brightest near the source. The light from a point light diminishes with distance. Look at the light in Figure 1.28. Notice that the light gives off a strong highlight on the ball. The area around the light is brightest; the areas on the corners of the plane are darker.

Point lights are good for lighting a room or a local area outside where artificial light is used (think a streetlamp at night). If you are creating a game that takes place primarily inside, much of the game will be illuminated with point light or similar lighting because it simulates the lighting of a light bulb.

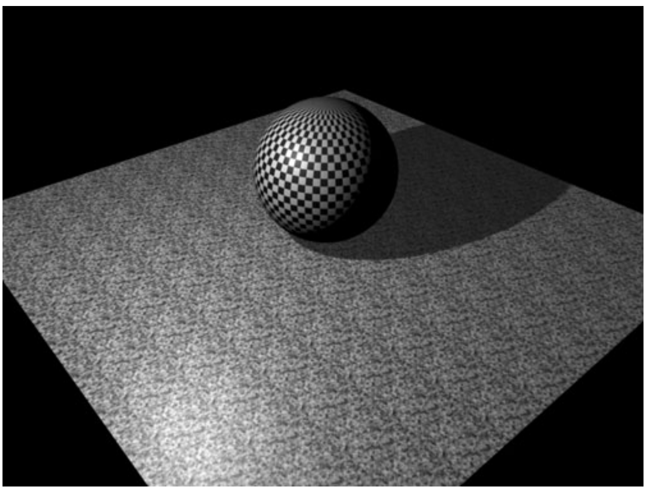


Figure 1.28
A point light is used to light this scene.

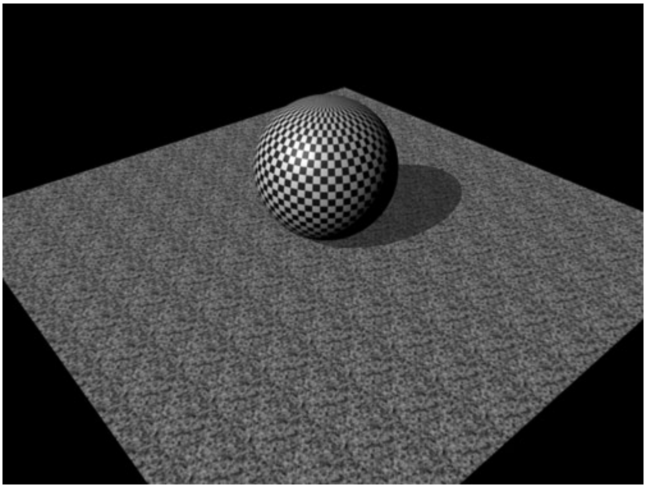


Figure 1.29
A directional light is used to light this scene.

Directional Light *Directional lights* simulate the light of the sun. The sun is a bright light source that is millions of miles away. Its light rays are almost exactly parallel to each other as they hit the earth's surface. A directional light illuminates everything evenly. Look at Figure 1.29, and notice how the light on the plane is even. There is no darkening with distance from the light.

Directional light is the most widely used light source in games because it is one of the easiest to calculate. It is also a light that people are comfortable with because they see it every day in real life. Directional light can be general, like in Figure 1.30, or can be limited, like a spotlight. Limited directional lights are called spotlights or conical lights.

Ambient Light An *ambient light* is a general light source that lights all surfaces in all directions. It is usually combined with a point or directional light to give a scene a more natural appearance. Ambient light is the result of light bouncing off one surface and then another. Most light in a home that comes through the windows is ambient light, even when the sun is shining directly through the window. Because the nature of light is to bounce from one surface to another, shadows are not completely black. You might have noticed that the shadows in Figures 1.28 and 1.29 are almost completely black, which gives them a very unnatural look. In real life, light bounces off everything, so almost all shadows have *some* light. Look at Figure 1.30 and notice that the

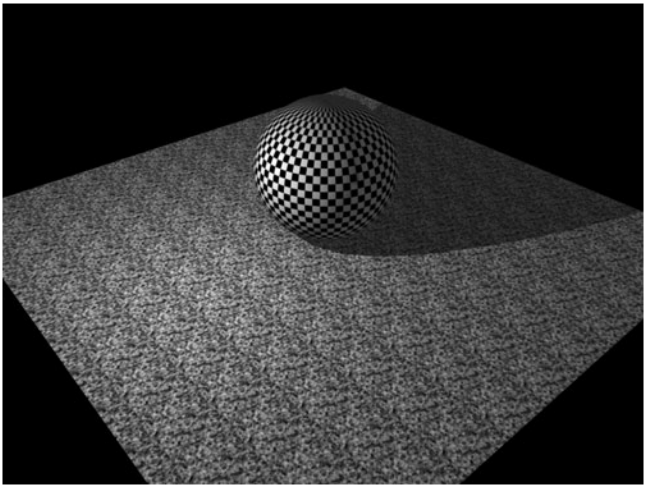


Figure 1.30
An ambient light is used to light the shadow areas.

shadows are much softer and the ball looks three-dimensional. The ambient light simulates the reflected light normally seen in real life.

Almost every scene needs to have some ambient light to make it look realistic. The only exception is a game set in deep space, where there is little or no reflected light.

Colored Light A *colored light* is not really a light; rather, it is an attribute of all the lights I have already discussed. Colored light is any light source that has a specific color. The scene in Figure 1.31 is lit with four separate colored lights. They are all point lights, but they each have a different color.

Almost every light source is a colored light. A fluorescent light is usually slightly blue, whereas a normal incandescent light bulb is yellow. The rays of the sun contain the full spectrum of light, but they tend to be a little on the yellow side—unless the sun is setting, in which case they can be very red and orange. When you are lighting a scene, use a light that fits the nature of the scene. For example, when you are doing an interior room, give it a slightly yellow light to simulate the incandescent light source.

If your game is being lit using a dominant colored light, you need to think about how that will affect your game character. For example, look at Figure 1.32. On the left, a white light is used to illuminate the head. On the right, a blue light is used. Notice that the blue light makes the colors of the face seem duller—almost as if

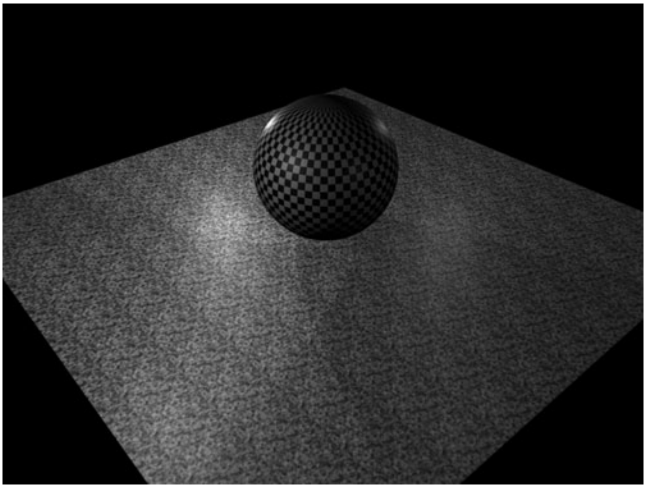


Figure 1.31
Four colored lights are used to light this scene.

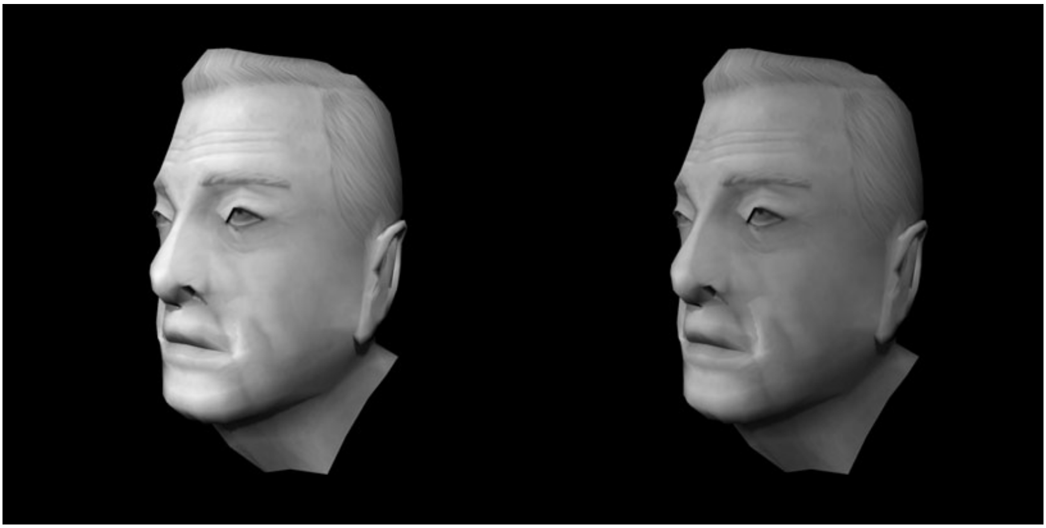


Figure 1.32
Blue light dulls the oranges of the face.

the character were a corpse. That's because blue is a complementary color to orange, meaning that it sits opposite orange on the color wheel. Colored light tends to dull its complementary color. Thus, if your game is using red light, it will dull the greens. If it is using orange firelight, it will dull the blues. Only white light will give the colors in your character full vibrancy.

Reflections

A shiny surface looks shiny because it reflects its surroundings. The only way to get a true reflection on a surface in a game is to calculate it based on the angle of the surroundings and the game view, and then render the calculated image on the surface. This usually has to be done for each ray of light; the process is called *ray tracing*. Ray tracing is impractical for many real-time game rendering because it takes up too much processor time. Some of the newer games on very high-powered systems use ray tracing in a limited form, but the process for ray tracing is so processor-intensive that it can only be used in special instances. In the absence of true reflections, game developers have come up with a few tricks that simulate reflective surfaces.

Specularity *Specularity* is a surface property that helps define an object as shiny or dull. Shiny objects have well-defined highlights. These highlights are direct reflections of the light source. By placing a well-defined highlight on an object, you generally give the object a shiny appearance—even without a complete reflection. Figure 1.33 shows two balls. The one on the right looks shinny because it has specularity as part of the surface qualities, while the one on the left does not.

Environment Maps Some surfaces need more than just a specular highlight to look right. A shiny car, for example, needs to reflect its surroundings. One way developers have created the look of reflections without resorting to the time-consuming process of ray tracing is by using an environment map. An

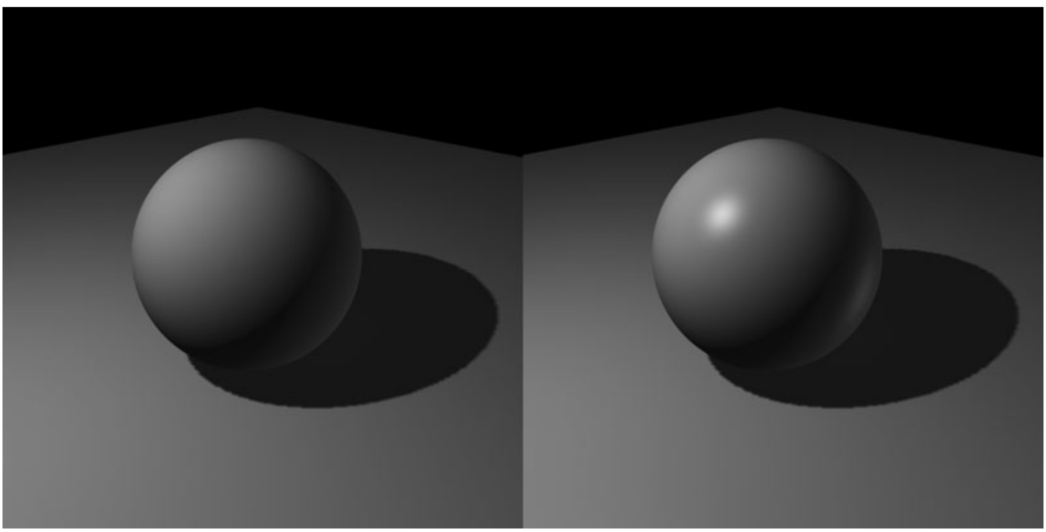


Figure 1.33
Different surfaces give each ball a different look.

Figure 1.34
An environment map.

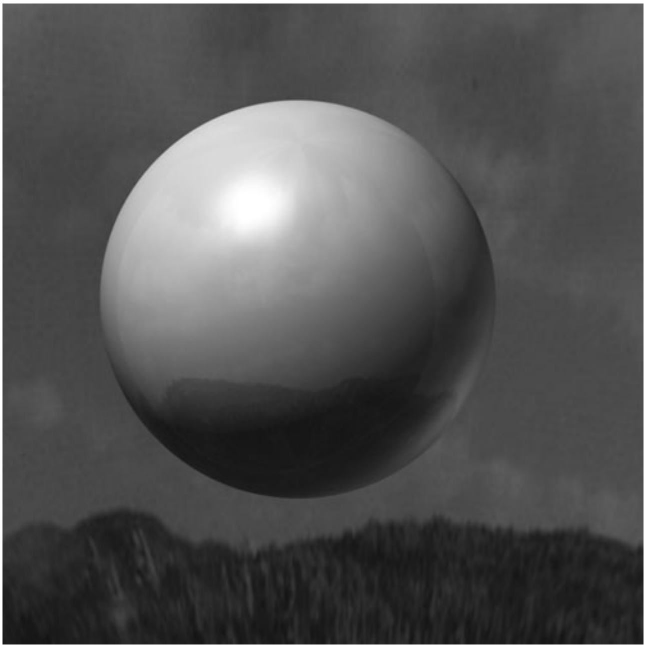


Figure 1.35
The environment map gives a close approximation of a real reflection.

environment map is a texture that gets rendered onto a surface as a reflection. The texture is usually a close approximation of the environment the object is in, and it can be changed from time to time as an object moves. Figure 1.34 shows an example of an environment map.

When the environment map is applied to the scene, it adds a reflective quality to the elements without the processing required for real reflections. Figure 1.35 shows a ball with an environment map applied to the scene.

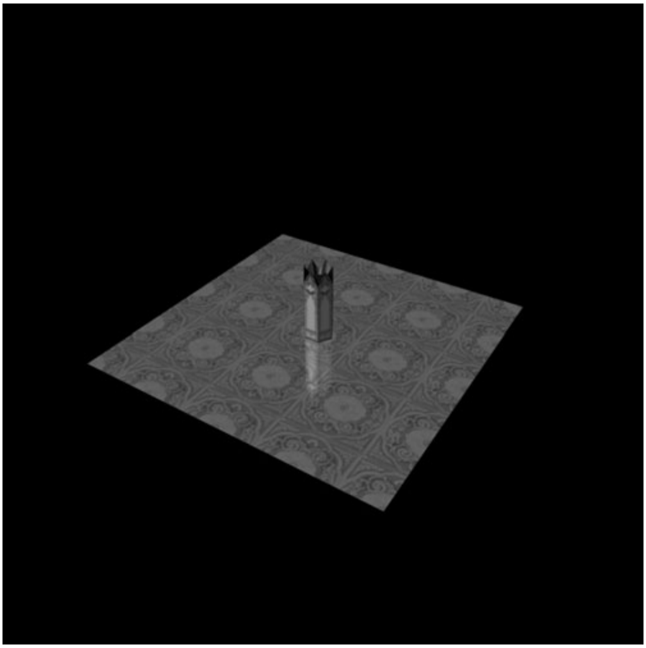


Figure 1.36
Games sometimes use actual geometry for reflections.

Rendered Reflections Another trick for achieving reflective qualities in games is to build the reflection geometry. This method works only on flat surfaces, such as floors, but it is very effective in giving a scene a realistic reflection. Take a look at how this is done. The lamp shown in Figure 1.36 is sitting on a tile floor. There is an exact duplicate of the lamp directly below the lamp, but upside down. The floor is semitransparent, so the lower model shows through the floor. The effect makes the floor look reflective and the objects move completely accurately for the reflection. This process obviously takes more polygons to create, but the effect is sometimes worth the extra geometry.

Animation

Before 3D graphics became popular in games, all animation had to be drawn one frame at a time in 2D. With the advent of 3D graphics, a whole new world opened for animators. Now they could create animation in a 3D environment and view the animation from any angle. Animations became separate elements from the art, making it possible to do one animation and apply it to multiple characters. Figure 1.37 shows a 3D character animated.

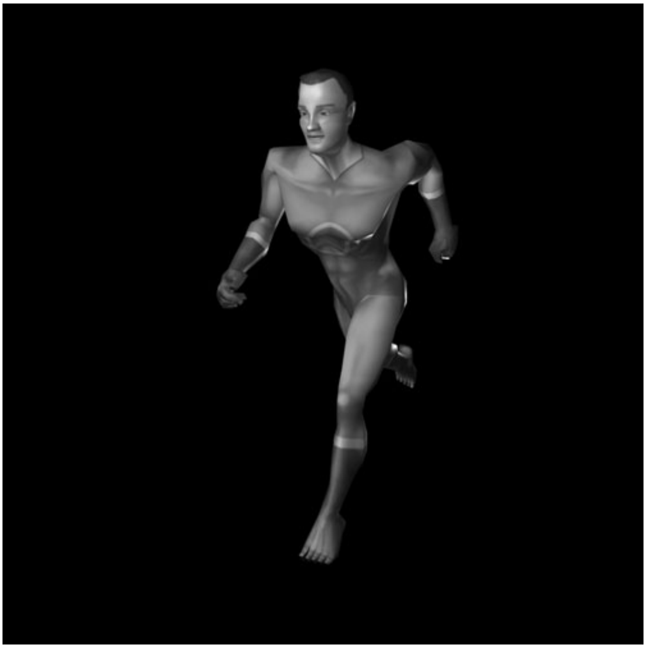


Figure 1.37
An animated game character.

Unlike 2D animation stored as picture files in a game, 3D graphics are stored as motion files, or they can be auto-articulated using software programs to drive the movement. Motion files can contain data on almost every attribute of a 3D model, including translation, rotation, size, color, and many others. There are basically two types of animation used in games: object animation and bone animation.

Object Animation

Object animation is basically moving a 3D object model from one location to another. It is the simplest of all animations. All object animation requires from the game system is to compute the object's location, rotation, and size from frame to frame. Object animation is sometimes done by the game programmers, but in many cases the artist will set up the animation in 3ds Max beforehand. Figure 1.38 shows a ball in flight or, in other words, object animation.

Bone Animation

Bone animation, or *joint animation,* as it is sometimes called, is the process of creating an internal object or character animation based on a system of bones

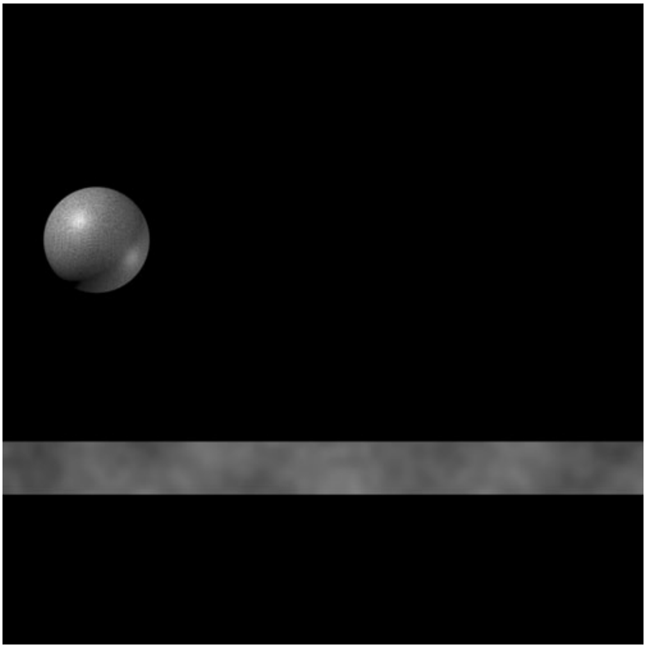


Figure 1.38
The ball moving across the screen is object animation.

and joints that make up a skeleton. Bone animation is by far the most complex of the basic types of animation and is used for animating characters in games. Figure 1.39 shows bones being placed inside a game character.

When setting up bone animation, the artist creates a skeleton for the model and then attaches the vertices of the model to the skeleton in a process called *rigging*. When a bone is pivoted on a joint, the attached vertices move with the bone. Vertices can often have more than one influencing bone. When more than one bone influences a vertex, it divides the movement between the two or more bones based on the percentage of influence from each bone.

Game AI and Blind Data

Game *AI,* or *artificial intelligence,* can mean a lot of things in game development, but for this chapter, I will simply use it to refer to the game code that controls where and how environments, objects, and characters are displayed. In other words, games are controlled by a complex system of software. This software determines what the player sees at any given time during the game. It also reacts to any input from the player.

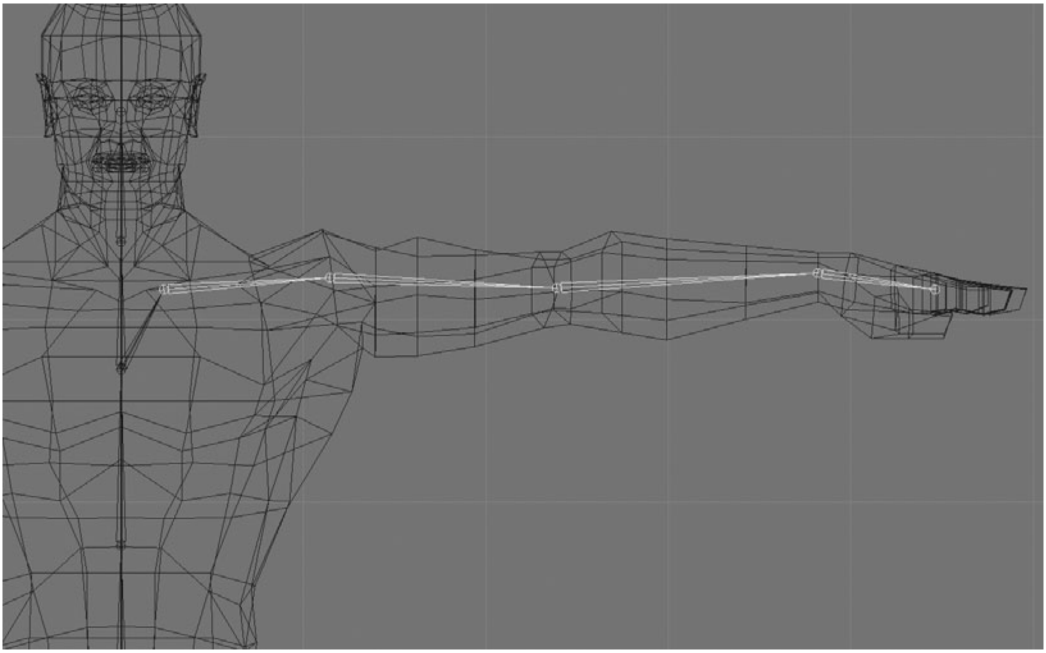


Figure 1.39
Bones are placed inside the game character.

As characters become more and more complex, the game artist will be required to give specific information to the programmers so they can create the special effects or special animation that a character may need. Sometimes, all the game artist needs to do is follow a naming convention, but other times he or she will need to include special unseen data with the model. In Maya, this data is called *blind data*. Blind data contains programming data used by developers to identify characters and to indicate any special things the character may need to do. Blind data can be applied to any part of a character so if the character is to hold an object, the blind data can indicate a location on the palm of the hand where the object is to be placed. Blind data is a very powerful tool for the character artist when setting up characters for games.

Bringing Everything Together

A lot things have to work together perfectly for even a single frame of a 3D game to be displayed. To get some perspective, think of it in these terms. A single setting in a game contains as many as 70,000 polygons. Each of these polygons has at least three vertices. Even if many of these vertices are shared, there could be as many as 100,000 vertices—or, in other words, specific virtual space locations. All of these virtual space locations have to be tracked by the game system not just once but several times every second. Many of the most popular games run at 60 frames per second. That means that in one second of game play, the system is tracking six million vertices. But it doesn't stop there.

In addition to simply keeping track of all of these vertices, the system has to apply textures to each polygon. These textures have to be mapped to the polygons so that they stay in place. Every polygon in the game has either a color or a texture applied to it. Sometimes these textures have more information than just a picture. They might have multiple layers containing information about transparency, bump, specularity, diffusion, and other attributes. So the game is tracking complex textures on 70,000 polygons 60 times every second. But it doesn't end there either.

Everything in the game has to react to light. Every polygon not only needs to have a texture, it has to have lighting. Many games have multiple light sources and dynamic lighting that moves during the game. So not only does the game system have to track lighting on a polygon from one light source, it also has to calculate the effect of multiple light sources on all polygons. But even then, we still are not done.

At any given time in the course of the game, there is a significant chance that the view will change and elements in the view will be in motion. In other words, the game system has to track and calculate the new positions of all the vertices, polygons, textures, and lighting on every frame. In addition, some elements are animating or moving independently of the view. Not only does the game system have to track where the moving objects are within the view, it also has to actually calculate those movements via the game AI as it reacts to whatever the player's input might be. Are you still with me?

All these game calculations have to be completed 60 times every second. Once everything is calculated, the scene can then be rendered to the display. Okay, let's think about the display, which is made up of individual pixels. During the rendering process, every individual pixel has to be given information on brightness and color. If a game is running at 1024 × 768 resolution, it means that for every frame, brightness and color information has to be delivered to 786,432 individual pixels. That is more than 47 million deliveries every second of the game.

Are you beginning to see why 3D games are so complex? Can you see why a very powerful computer system is required to run 3D games?

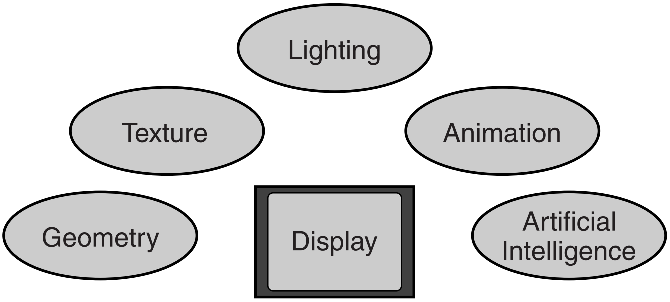


Figure 1.40
Displaying a 3D game takes many steps.

Why is this so important to the artist? Well, understanding what it takes to display even a single frame of a 3D game will help you to plan and execute your art in such a way that you will be able to get the most out of the game you are creating. Because even the most powerful game systems have a limit on the number of calculations they can process at any given time, you have to set limits on the number of polygons, textures, lights, and animation processes. Sometimes these limits may vary. For example, you might have a need for a lot of specialized lights in a particular part of your game. To keep the frame running smoothly, you can reduce the number of polygons and textures in that area. Figure 1.40 shows the steps in the process of displaying a 3D game.

I think it is important for you to understand that this is only an overview of what it takes to run a 3D game. The entire process is actually much more complex. I haven't even covered things such as particle effects and physics models. But the purpose of this chapter is to help you get a feel for the complexity and overall scope of creating a 3D game so that when you create a game character you have some background in why there are limitations on geometry, texture size, and number of bones used.

Summary

This chapter has been an overview of character development and 3D art as it relates to games. This chapter covered several important points:

- Game characters
- Types of game characters
- Displaying art in games

- The history of computer art and game characters
- 2D and 3D art programs
- Components of 3D models
- Textures on 3D models
- Lighting of 3D models
- Animation of 3D models
- Rendering 3D models to pixels

The programs used in this book are contained on the CD that accompanies the book. The programs include the following:

- Corel Painter
- CorelDRAW Suite
- Jasc Paint Shop Pro
- Maya PLE

Now that you have finished this chapter, you are ready to move forward and start to create great game art. Your only limitations will be those imposed by the game system and your own knowledge and talent. As you grow in your abilities, you will find a wonderful and rewarding world of creative expression. Your opportunities are limitless. Welcome to the world of character development for games.

Chapter 2

Developing Character Ideas

Creating good game-character designs is one of the most satisfying and fun aspects of game art. Like the actors and actresses in a motion picture, they are the focus of attention. They are the performers on the stage of the game. Often, the player is connected to the game through the game characters. Thus, creating good character art is very important. Figure 2.1 shows a character that we will be creating as an example in this book.

Characters in Games

With the possible exception of some puzzle games, almost every game has characters. A *game character* doesn't have to be a person or creature. It can be any intelligent person, creature, or machine in a game. If you want to make a game where a stick is the main character, you can go right ahead. The important thing is not what the character is, but rather that the character has some intelligence.

Game characters include all intelligent people, creatures, or objects in a game. *Intelligent* means that the character is either controlled by the player or by the game software. The specific software that controls characters in games is called *artificial intelligence*, or *AI* for short. AI is used in a game to have characters act or react to game situations so the world feels like it is a plausible world. AI in games is becoming increasingly complex to the point that the player can interact with AI-controlled characters as if they were characters controlled by other human players.

Figure 2.1
Here is a picture of the character that you will build in this book.

When game characters first started appearing in games, they were only vague representations of people or creatures. There isn't a lot that an artist can do when the game character is only 16 pixels high. Advances in technology have led to an increase in character complexity. Indeed, characters in games are becoming more and more lifelike to the point that they don't just run around and shoot things; they can also have emotion. An angry character will not just attack with sword drawn, he will have facial expressions and body language to emphasize the attack. If the game design calls for it, characters in games can have a full range of emotions. When designing a game character today, the artist must think in terms of a complete character accounting for personality and tendencies as well as looks.

Character designs start with character ideas. Ideas are usually written out in character descriptions generated by the game designer and given to the game artist to visualize. The game artist works with the game designer to create sketches from which the 3D models are derived. The steps for visualizing and creating game characters are as follows (see Figure 2.2):

1. Character description
2. Thumbnail sketches
3. Character sketches

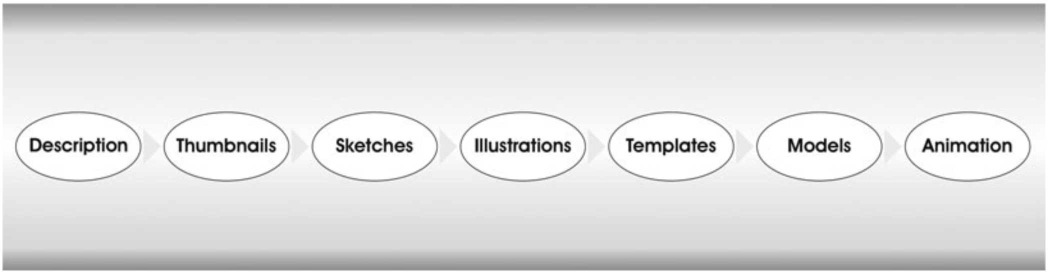


Figure 2.2
There are several steps to completing a game character.

4. Character illustrations
5. Character templates
6. Character models
7. Animation

This chapter covers the process up to character sketches. Chapter 3, "Character Illustrations and Templates," covers the next two phases of character design. In subsequent chapters, character models and animation are covered.

Character Descriptions

A *character description* is a written description of the character. Depending on the type of game and the importance of the character, the description can be just a few sentences or it can be several pages long. Whether it is short or long, the important aspect of the description is that is gives the game artist a clear image of the character, and gives the script writer a good understanding of the character's personality. Game characters are getting more and more sophisticated, so simple descriptions of what the character looks like are no longer enough.

A good character description needs to have several elements to fully describe the character. These elements include the following:

- **Type.** Most games use human or human-like characters. This is probably due to the fact that human-like characters are easier for the player to identify with than non-human characters. In some games, like sports games, the characters are very human; often, they need to look so realistic that they are hard to distinguish from real people. And given that the trend in video games is more realism, even fantasy characters need to look like they really could be a living breathing person. That said, not all characters in a game are human or even human like. Some characters are animals, as is the case with the many pet games. Some characters are fantasy creatures or science-fiction entities. In some games, the characters may have nothing at all to do with human anatomy. For example, in a futuristic science-fiction game, the characters may be robots. The only limit on the type of character in a game is the imagination of the game designer.

- **Size.** It is important for the game artist to know the size of the character in relation to the other characters and the game setting. A giant has a very different build and carriage from a wood nymph. A football player looks very different from a figure skater.

- **Gender.** I think it goes without saying that the game artist needs to know if the character is male, female, or indeterminate. After all, the gender of a character can play a huge role in how a character is perceived by the player. It also plays a big role in a character's physical nature. Feminine characters are usually smaller, sleeker, and more elegant in their movements, while male characters tend to be larger, bulkier, and more forceful. (This isn't to say, however, that both genders can't share these qualities or break these rules.) If a character is of indeterminate gender—for example, a worm or energy entity—the designer should communicate to the artist the nature of the character; without the clue of a gender, it may be harder for the artist to visualize the attributes of the character. In addition to indicating the gender of the character, be it male, female, or indeterminate, the designer should also emphasize how the character feels about his/her/its gender. A female character, for example, may be very feminine in the way she dresses or moves, yet still have attributes generally associated with male characters such as aggressiveness and physical strength. Sometimes it is the juxtaposition of sexual characteristics that make the character interesting.

- **Race.** Race is an important part of the description of a human character because it deals with skin tones, which can have a dramatic impact on a character's appearance. In addition, although race can mostly be a visual aspect of human characters, it also tends to have more of an influence on the behavior of fantasy and science-fiction characters. This is mostly due to the fact that these characters tend to involve exaggerated racial stereotypes.

- **Dress.** Many characters wear some sort of clothing. It may be as simple as a loin cloth or as ornate and exotic as a kimono. In real life, we often define ourselves by what we wear. Words like punk, gothic, business casual, and formal all bring up images of attire. The type of clothing a character wears can be very influential in defining the character. Another aspect of dress is body armor. Many characters use clothing as a shield. Sometimes it is obvious, such as with a medieval knight; other times it is more subtle, as with a girl who wears loose clothing to disguise her gender. Armor can be physical or psychological. It can be obvious or hidden. When a character wears armor, it often changes the nature of the character's behavior. Characters who feel protected often have a greater sense of confidence.

- **Power.** Power is the ability to act. All characters have some power. A character without power is not really a character at all; rather, it is an object. When defining a character, the designer should indicate those powers that are unique to the character like strength, intellect, wisdom, flight, and dexterity. Power affects how a character interacts with the game world and with other characters. For example, a game character might be strong but also heavy, while another character might be very light yet not very strong. In the game world, the strong, heavy character would move more slowly and more ponderously, and the light, weaker character would be able to move swiftly and might also be able to climb or jump well. These characteristics need to be communicated to the game artist who will be designing the characters.

- **Skill.** In many ways, skill is similar to power, but skill is more of a learned characteristic while power is more of an intrinsic characteristic. Skills can range from magic to bead making. In some games, such as role-playing games, the characters often grow in skill during the course of the game. In other games the character's skills are set. For example, a lineman in a football game already comes with the skill for blocking defenders. A driver in a racing game already knows how to drive. In some games, skill is based on player input. This is often the case with fighting games where combo moves are common. In these types of games, the character's skill is equal only to the player's ability to play the game.

- **Personality.** Game characters need to have personality to be recognizable in today's competitive marketplace. They need to convey to the player a sense of presence. That means the character needs to be cast, similar to the way a character is cast a motion picture, but instead of looking for an actor to play the part, the game designer designs the character's personality. Personality can greatly affect how a character both looks and acts in a game. A shy character will move very differently in crowded environments than a boisterous, loud-mouthed character. Personality traits can include things like confidence, swagger, shyness, intolerance, and charisma, to name a few.

Note

Personality is more important for AI controlled characters than for player controlled characters because the player's character will often take on the personality of the player. This is especially true for role playing games where the player designs the character from the ground up.

- **Accessories.** Often, characters carry items with them. Sometimes the item may be a walking staff; other times it might be a weapon or even an object of power. Some games, such as role-playing games, use accessories extensively, creating elaborate systems for inventorying them. Adventure games are often about finding items that the player will be able to use later. Some adventure games use accessories to allow progress in the game, like keys to unlock doors or messages to help the player through the game.
- **Relationships.** Relationships may not directly affect how a character looks (other than that they should resemble their mother and father), but it can have a big impact on how the character acts. A mother will tend to want to protect her children. A slave master will treat slaves differently from free people. If the hero loves the maiden in distress, he will be more motivated to help her than if he doesn't know her. Relationships can also affect the progression of a game. For example, a character may need to make friends with the local police to help with a quest. The police will then help the player in finding someone who can lead the player to the evil villain's lair.
- **Social status.** A character's social status affects both the way the character looks as well as the way the character acts. Often, social status is indicated by clothing or some other marking. For example, the enemy might wear uniforms that show their rank. Some games have complete social systems with leaders and followers. Social status can also affect the player's character. Players may have to gain in social status in order to complete quests or compete in higher-level tournaments. For example, a racing game might

require the player to win a certain number of races to move from amateur to circuit races.

This list may not include everything that a particular character might need in its description, but it does include the basic information necessary for the game artist and writer to create the character.

Writing Character Descriptions

Now that you know some of the basics of writing character descriptions, it is time to write a few. Following is an example of a character description from a game.

Boric
Boric is a 35-year-old Haladin male ranger from the northern frontier. He is muscular, 6 feet and 1 inch tall, with a sun-browned complexion. His hair and beard are black. His eyes are brown. He wears mostly hardened leather armor with the exception of a plain steel breastplate and horned steel helmet. His favorite weapon is the longbow, but he is also proficient with the sword and carries one strapped to his waist. He also has a long knife in a scabbard on each boot.

Boric was born in southern Haladin and moved north with his family when he was 11 years old. His father was a game warden for the king, and the family lived in a small cottage near the northern game reserve. It is there that Boric first gained his love for the forest and the beauty of the northern wilderness.

Boric trained under his father in woodcraft and the use of weapons until he reached the age of 17, when he left home with other young men of the area to fight for Haladin against the tribes of the northern steppes who were invading northern Haladin at that time. For the next 10 years, Boric served as a military scout under Duke Walrinic, who had command of the king's northern armies. Boric distinguished himself well as a scout and returned home after the 10-year war with the northern tribes with the rank of Master Scout.

For the next three years, Boric worked with his father on the game reserve but, missing northern frontier, he left home at the age of 30 to travel to the northern frontier and serve once again under Duke Walrinic as a ranger spying on the movements of the northern tribes.

> Boric is a hardened veteran of many battles and skilled in the use of many weapons. He is especially good with the longbow and won several competitions with his fellow rangers. He is a master of concealment and can move silently and effortlessly through the northern wilds.
>
> Boric has a pragmatic, practical personality, and is quiet by nature. He never married because he didn't deem life as a ranger very good for raising a family. He spends his free time studying nature and has gained the use of some minor Druid spells that he uses to enhance his scouting abilities.

Does this character description give you a good idea of the detail that might go into creating a character for a game? As you read the description, were you able to visualize the character? Try writing a few character descriptions yourself. See if you can create a few of your own characters. Keep in mind the information that should go in a character description.

Drawing

Once you have a detailed character description, the next step is to draw the character. If you are new to drawing, creating a picture of your character might be a little intimidating. With a little help and understanding of basic drawing principles, however, the process can be a lot easier. Almost anyone can learn to draw if they work at it.

Note

> If you already have a basic understanding of drawing, you can skip this section and go directly to the one titled "Character Thumbnails."

If you are new to drawing or haven't had a lot of training, you will likely have learned a few bad habits that you will need to break. Some of these habits include the following:

- Holding the pencil incorrectly
- Drawing too darkly in the beginning stages of the drawing
- Not blocking in the general shapes before adding detail
- Erasing too much

Let's start by looking at how to hold the pencil.

Holding and Using the Pencil Incorrectly

One of the biggest problems beginning artists have is that they don't hold the pencil correctly for drawing. The problem is not so much that the artist holds the pencil wrong as it is that the artist needs to learn how to hold the pencil in different ways for different purposes. Most beginning artists will hold the pencil for drawing the same way they hold it for writing.

Drawing is a very different process from writing. When a person writes, he or she is creating precise, small characters on a piece of paper. Although this action is similar to drawing where precise detail is needed, it is very different from the action needed for blocking in a drawing or for achieving fine, smooth shading.

When a pencil is held, beginning artists tend to tighten up and limit the movement of the pencil to the range of motion in the fingers. This is because the hand rests on the paper. Figure 2.3 shows the typical writing position of the hand.

In the initial stages of a drawing, freedom of movement is very important. To free your hand to move about, try holding the pencil similarly to how you would hold a butter knife. Figure 2.4 shows how the pencil should be held.

Holding the pencil in the butter-knife position makes it almost impossible to draw with only finger movement. Instead, the whole arm is brought into play. If this is the first time you have ever drawn holding a pencil in the butter-knife position, it may feel a little awkward at first. Stay with it until you become

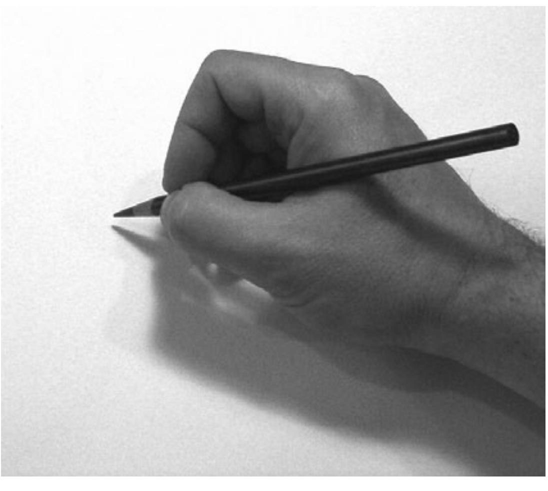


Figure 2.3
When writing, the palm of the hand rests on the paper.

Figure 2.4
Hold the pencil like you would a butter knife.

comfortable. The long-term benefits from learning to hold your pencil in the butter-knife position are immeasurable. It will allow you to draw with more freedom than the writing position.

Let's do a little practicing. Find yourself some paper and a pencil. The paper should be large because I want you to have to move your arm. The pencil should be a soft pencil so that it will be easy for you to see your drawing. I recommend using newsprint for the paper and at least a 2B or softer pencil.

Figure 2.5 is a picture taken from my book *The Animator's Reference Book,* published by Thomson Course Technology. This book is a great source for artistic reference of the human body in a variety of actions. I will use it here to help you loosen the way you start a drawing.

Prop your drawing surface up so that it is almost parallel with your own torso. This will make it easier to draw with the pencil in the butter knife position. Now quickly sketch the photo. Try to capture the essence of the figure's action in less than 60 seconds. Yes, this is speed drawing and it is good for you, so don't complain about the time limit. You're not doing a finished drawing. This is just an exercise. It's kind of like stretching before a long-distance race. You need to get loosened up.

Figure 2.6 is a sketch I made of Figure 2.5. Your sketch should be similar, but it is okay if it is not exactly the same. I made this sketch in about 30 seconds, focusing only on trying to capture the feeling of movement. There probably isn't a single

Figure 2.5
Draw this figure quickly and loosely.

Figure 2.6
Envision the skeleton of the figure when drawing.

line in the drawing that is technically accurate with regard to the exact outline of the figure. In fact, most of the lines in this drawing follow the skeleton of the figure more than they do the outside shape.

In order for you to draw with the pencil in the butter-knife position, you have to move your arm, not your fingers. The human body is an organic yet mechanical mechanism. The muscles in our arms are like hydraulic pistons, contracting and expanding to create movement. Unlike hydraulics, however, the placement of the muscles and the way they are attached to the bones of our skeletons is organic. This is very important because it allows the body to move in fluid, non-mechanical ways. It is the attachment of the muscles and the shape of our bones that allow for this grace of movement. Why is graceful movement important? It's because in many ways, the artist needs to learn how to dance with the drawing.

Dancing With the Drawing

Dancing with the drawing means that the artist takes advantage of the natural motions of the body to add expression to the drawings. By holding the pencil in the butter-knife position, the artist is better able to dance with the drawing. In other words, think of yourself as a dancer. Try to move your arm in graceful, smooth, flowing motions. It isn't that you are giving up control of your drawing. It is rather that you are gaining control by letting the natural rhythm of your body express itself in your drawings.

Now try a couple more drawings. Figures 2.7 and 2.8 are two more photographs from *The Animator's Reference Book*. Quickly sketch both pictures. Try to just indicate as quickly as you can the basic motion captured in the photo. If you need to, draw them several times until you feel good about your sketches. The idea here is not to create an accurate picture but rather to indicate the essence of the figure and the actions.

Hint

As you draw, you will hopefully notice that there is a main line of motion in each figure. The main line of motion is the foundation line of capturing the motion of the figure. To find the main line of motion, create a line from the head, following the spine, and down through the leg that is holding most of the weight of the body. Try drawing this line first and then attaching the rest of the body to this line. It will help you to gain a feeling for motion.

The butter-knife position is great for starting a drawing but it doesn't work to well for fine detail of delicate shading. The writing position is a better for fine

Figure 2.7
The figure is walking while holding a sword.

detail, but you will need to learn another way to hold the pencil for detail and delicate shading.

Too Dark

Beginning artists will often draw too darkly in the beginning stages of the drawing. In the initial stages of drawing, the lines should be kept very light until the drawing is defined; then the darker lines and shading can be added. Drawing dark lines in the beginning of the drawing defines almost immovable lines on the paper. Very seldom does the artist get every line of a drawing correct on the first try. Therefore, lightly laying in the drawing in the beginning makes it much easier to adjust the drawing until it is exactly right. If the initial lines for the drawing are too dark, even a good eraser may not be able to remove them, as shown in Figure 2.9.

There aren't too many things more intimidating to the artist than a white sheet of paper. Often, the beginner is afraid to make a mark on the paper for fear of not getting it right. Drawing very lightly is less intimidating and helps to get past the fear of getting the drawing wrong because even if the first few lines are not correct, they are so light that it is very easy to erase them.

Figure 2.8
The figure is leaning forward while walking.

Figure 2.9
The dark line on the near the forehead causes problems.

Not Blocking in the Drawing

Most drawings look better it they fit on the paper. Often, beginning artists will start a drawing on one side of the paper and by the time they get to the other side they run out of room. If you ever tried to draw a game character and ran out of room for his feet, you know what I mean. Not blocking in the drawing is usually the reason why the beginner has trouble fitting the drawing on the paper.

Blocking in a drawing is the first step in defining what the drawing will become. During this stage, the artist uses lightly drawn lines to define the elements of the drawing. The composition is determined, the major shapes are defined, and the proportions of each element in the picture are delineated. It is basically like creating a plan for the drawing. Figure 2.10 shows a drawing of some fruit. The drawing on the top is the drawing blocked in while the one of the bottom is the shaded drawing.

With few exceptions, drawings can be broken down into basic shapes. Defining elements of a drawing into basic shapes helps the artist understand the nature of each form. By understanding the nature of each form, the artist is better equipped to set up the drawing.

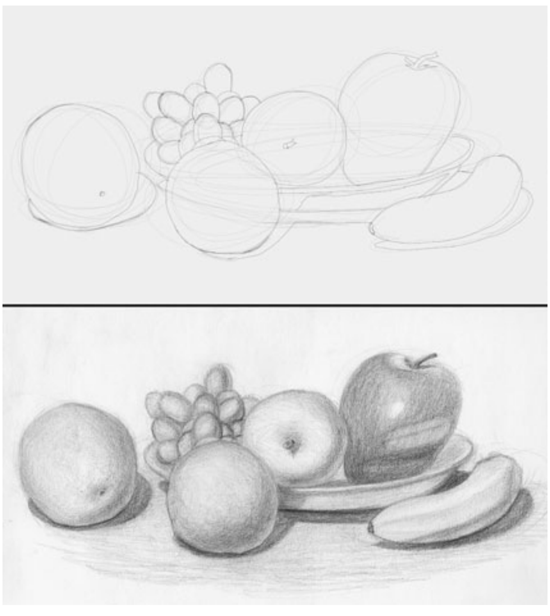


Figure 2.10
Blocking in the drawing helps to define proportions and size.

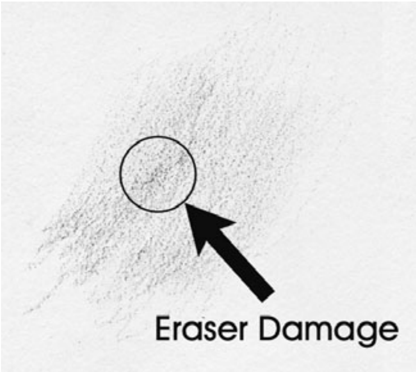


Figure 2.11
The eraser can damage the paper.

Too Much Erasing

Be good to your paper, and your paper will be good to you. Paper fibers are delicate and become damaged very easily. Harsh drawing and excessive erasing destroy the surface quality of the paper. Once the paper is damaged, it can't be repaired. Figure 2.11 shows eraser damage on a piece of paper.

Damaged paper creates an uneven drawing surface. The damaged area will receive the graphite differently from the undamaged areas. If the paper fibers are raised by the abrasive use of an eraser, the damaged area will have more tooth than the surrounding paper. It is almost impossible to compensate for damaged areas when doing delicate pencil work.

A common mistake of beginning artists is to use the eraser too frequently. Experienced artists rarely erase. If you must erase an area, the best eraser to use is a kneaded eraser. Kneaded erasers are less abrasive than other erasers and they leave less residue behind. They also can be molded to any shape—a useful trait when you want to do fine touchup work on your drawing.

Character Thumbnails

Thumbnail sketches are quick idea drawings used by artists to develop compositions for more detailed drawings. Often, an artist will sketch several ideas when working out a larger or more detailed drawing. Many artists, however, wish to jump right in and draw a finished drawing right away. The problem with going directly to the finished character drawing is that there is no plan; the drawing will suffer because not enough thought was put into the design. It is a much better

Figure 2.12
A thumbnail sketch of a cartoon character.

idea to work out the overall design in a thumbnail sketch. Even just a few quick sketches will help the artist to define and plan a good design for a final character drawing.

It is easy to change the design of the drawing at the thumbnail level because the drawings are quick and the artist hasn't made a big commitment of time of effort. If the design doesn't look good in the thumbnail sketch, the chances are that it will not look good in the finished drawing either. The more design elements that an artist can work out in loose thumbnail sketches, the better the finished drawing's design will be.

The nice thing about thumbnail-sketching characters is that the artist can follow his or her imagination. The sketches are first drawn loosely as the artist searches for an idea and then refined to give the artist a clear picture of the major features of the character. Figure 2.12 is a thumbnail of a cartoon character. The character is much shorter and wider than a normal human.

One way to design a unique character is to exaggerate the character's features. In the drawing in Figure 2.12, the woman's face is very wide with a large nose and glasses that are so powerful that her eyes appear twice their normal size. If you look closely, you will see some of the construction lines.

Figure 2.13 shows a more normally proportioned character but from a sharp perspective. Looking up at the character enhances his size, making him appear to be a giant.

Sometimes it is useful to put the character in a setting to help the artist see the character in context. A setting will help to define the character and give it scale.

Figure 2.13
This character is drawn with a sharp perspective.

The earlier sketches were of the character by itself. Figure 2.14 shows a character by a tree. The tree and the girl reading not only define scale, but also mood.

Drawing Characters

Drawing characters is maybe one of the most difficult tasks a game artist will have to deal with. The problem is compounded when the character is human. The issue with human characters is that everybody is familiar with the human form. If the artist moves a branch on a tree, for example, few people will notice. Most people will, however, notice if a person's eye is moved a fraction of an inch. Look at Figure 2.15 to see what happens when the eye is moved only a little.

Start With the Head

Game characters have all shapes and sizes of heads, from cartoon styles to very realistic. Using sound drawing principles will improve the design of the character no matter what the style might be. Let's start with a simple cartoon-style head and then move on to a more realistic approach. Begin with a circle, as shown in Figure 2.16.

Figure 2.14
Sometimes a character will be drawn within a setting.

Figure 2.15
Look for the difference in the eye placement of these two drawings.

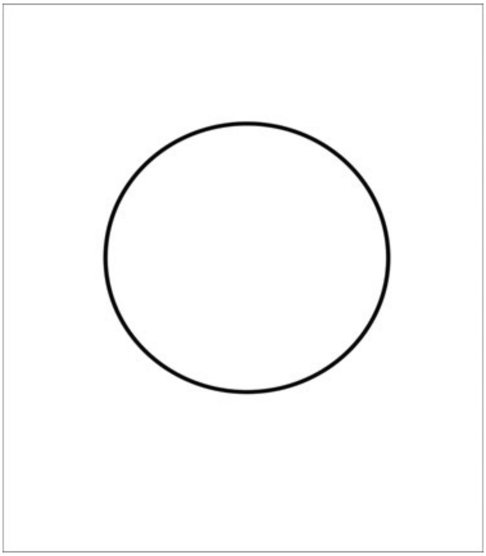


Figure 2.16
Begin drawing the head by drawing a circle.

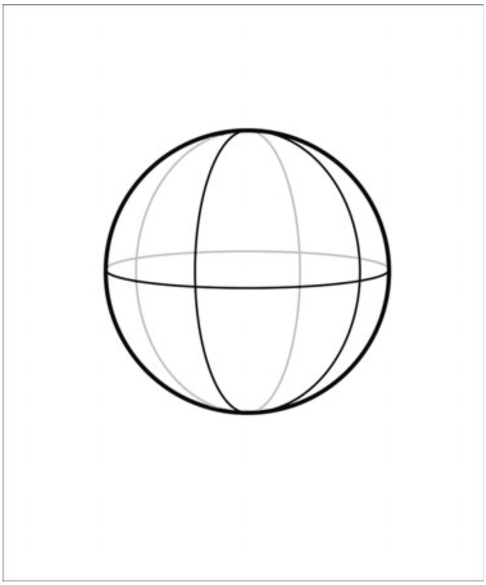


Figure 2.17
Draw some longitudinal and latitudinal lines around the circle.

The circle now needs to be made into something that has dimension. It needs to become a three-dimensional object. Drawing some additional lines around the circle to indicate a sphere, as shown in Figure 2.17, does this.

Figure 2.18
Use this picture to copy and practice drawing heads. (Photo-copy it first, so you can use it more than once.)

In the preceding example, the lines carry through the circle, but the lines on the front of the circle are bolder than the lines on the back to help avoid confusion. The lines not only give the sphere dimension, they also help with the placement of features. These lines are called *construction lines*. They are usually drawn very lightly. Figure 2.18 is a guide that you can copy from the book to practice drawing a cartoon head.

Figure 2.19
Lightly draw the features using the construction lines as guides.

Figure 2.20
Draw the character's head over the construction lines.

Use the lines on the sphere as a guide to place the features of the face. Where the two lines cross on the front of the sphere is where the character's nose is placed. Place the eyes along the horizontal line and the mouth and nose along the vertical line. Place the ears behind the other vertical line along the side of the face, as shown in Figure 2.19.

Once you have a good idea of where the features should go, then go ahead and draw the head in bolder lines (see Figure 2.20). Using this method will help to keep the features where they belong and give the face a feeling of dimensionality.

Realistic Proportions

In figure drawing, *proportion* is a term used to describe accurately defining relative distances between physical features of the human body. This means that when drawing people, all aspects of the body are related to each other so that no part is drawn too small or too large.

Proportions of the Head

Let's start looking at proportion with the head. Figure 2.21 is a drawing of a head from the front and the side.

While there are individual variations, most heads fall with in some general guidelines.

- The face can be divided vertically into four sections. The hairline is in the top section. The top of the eyes and eyebrows are in the next quarter section. The bottom of the eyes and the nose are in the next quarter and the mouth and chin are in the bottom quarter (see Figure 2.22).
- The distance between the eyes is about one eye-width, as shown in Figure 2.23. Notice that the nose is also about an eye-width wide.

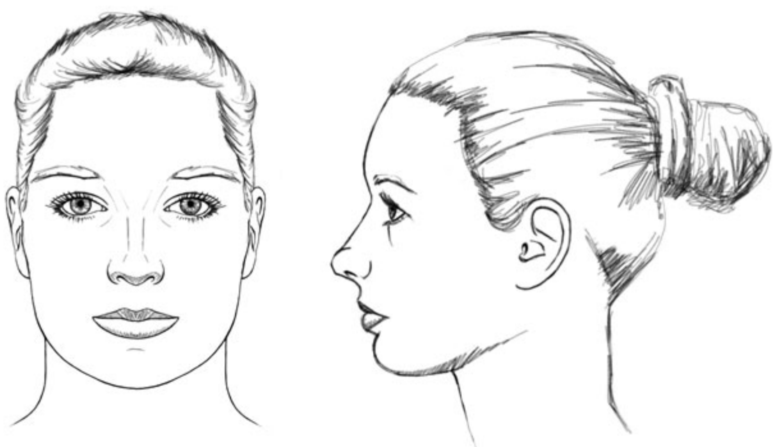


Figure 2.21
A head from the front and side

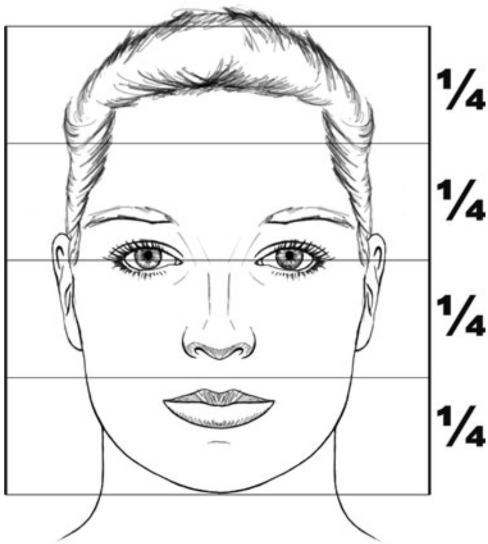


Figure 2.22
The face can be divided vertically into four quarters.

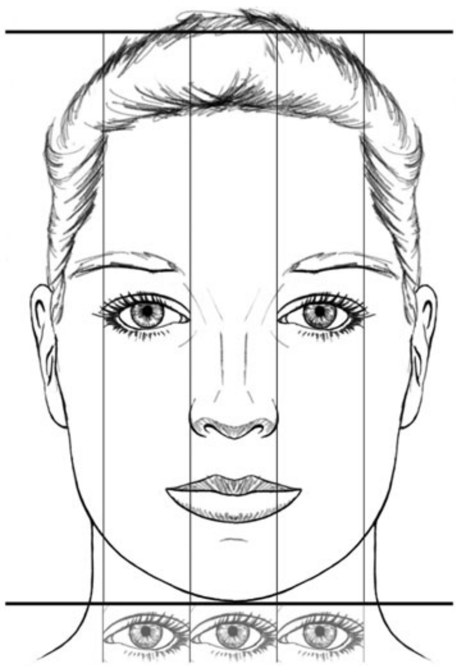


Figure 2.23
The eyes are about one eye-width apart.

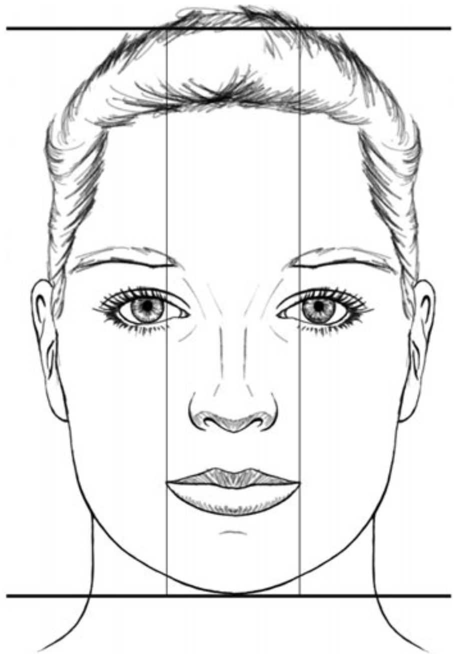


Figure 2.24
The mouth is usually inside the width between the pupils.

(Noses vary in width quite a bit, so this is not always the case—but it's a good guideline starting out.)

- The width of the mouth generally falls inside the distance between the pupils of the eyes, as shown in Figure 2.24.
- The ears usually are as high as the top of the eyes and extend to near the bottom of the nose, as shown in Figure 2.25.
- The ears usually fall in the back half of the head, as seen from the side (see Figure 2.26).
- When measuring from the tip of the nose to the back of the head, the base of the front of the neck falls at about the midpoint, as shown in Figure 2.27.

Construction Guide for the Head

With these principles in mind, you can now create a construction guide for drawing portraits. To help them accurately draw in 3D space, artists use construction guides as a framework for placing features. Here is how it works.

1. Start with a circle. The circle will act as a guide for most of the skull. The bottom of the circle will generally fall somewhere between the mouth and

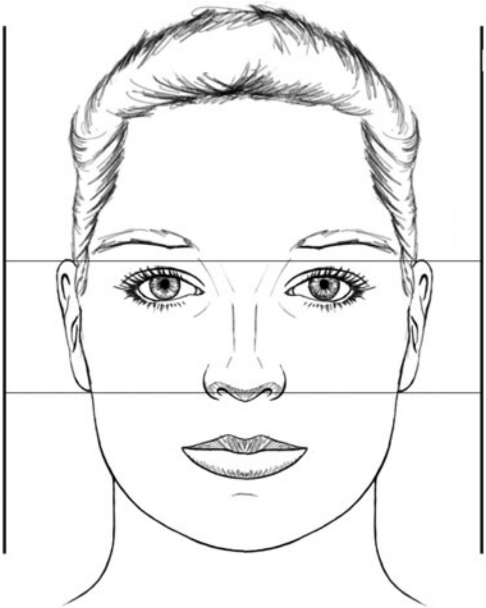


Figure 2.25
The ears are about the length from the top of the eyes and the bottom of the nose.

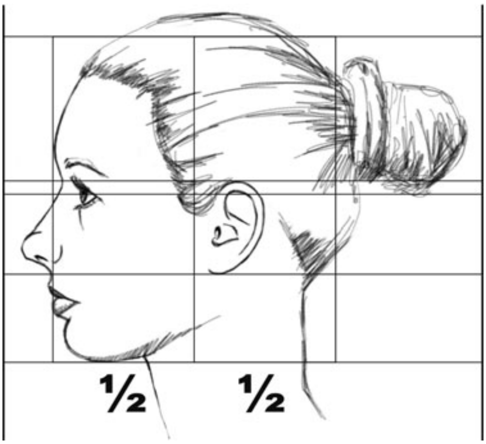


Figure 2.26
The ears are more than halfway back on the head.

the nose, and most of the time it comes just below the cheekbones, as shown in Figure 2.28.

Note

Notice that the circle extends out from the head on either side, past the ears. This is because the head is not a perfect circle. When drawing, you need to remember to bring in the sides of the head inside the circle.

Figure 2.27
The front of the neck starts about halfway back on the head.

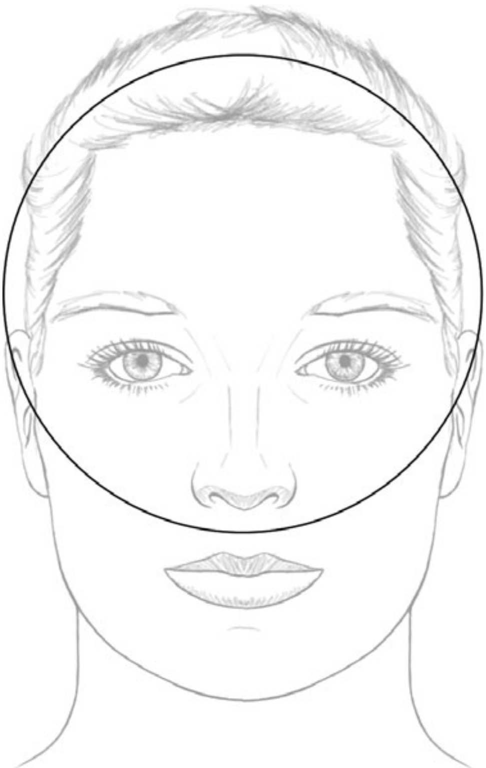


Figure 2.28
Start the drawing with a simple circle.

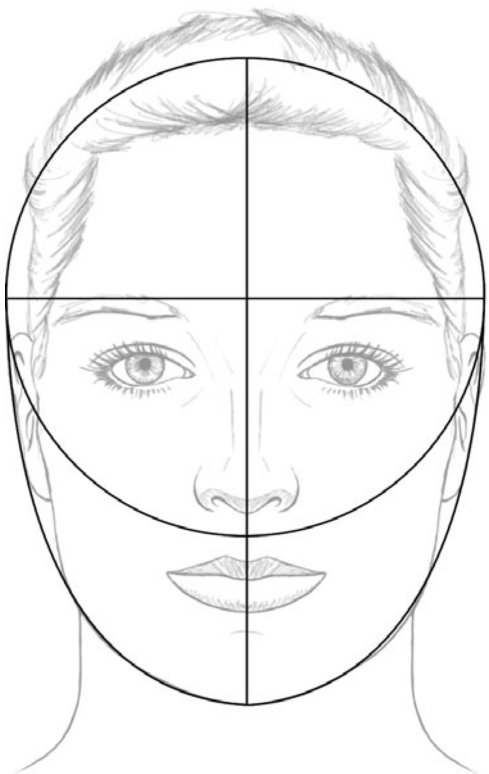


Figure 2.29
Draw in the jaw line.

2. Split the circle in half both vertically and horizontally. The head is fundamentally symmetrical. The vertical line defines the center of the face. The horizontal line is used as a guide for placing features.
3. Once you have established the circle, draw in the jaw line. It will extend down below the circle, as shown in Figure 2.29. Extend the vertical line to the bottom of the jaw.
4. Divide the head construction with horizontal lines for the eyes, nose, and mouth, as shown in Figure 2.30.

This is the basic construction guide for drawing a head from the front. Figure 2.31 shows the construction for drawing a head from the side.

Not every head you draw will be a front view or a side view. Often, the head will be turned to one side or the other, or will be looking up or down. Most of

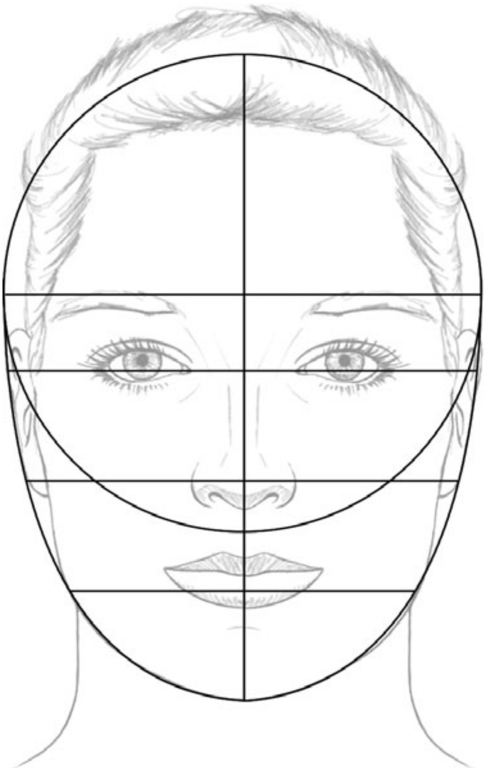


Figure 2.30
Draw lines for the eyes, nose and mouth.

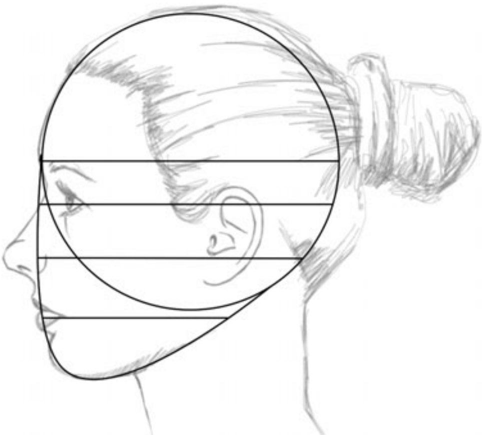


Figure 2.31
The construction lines work for a side view as well.

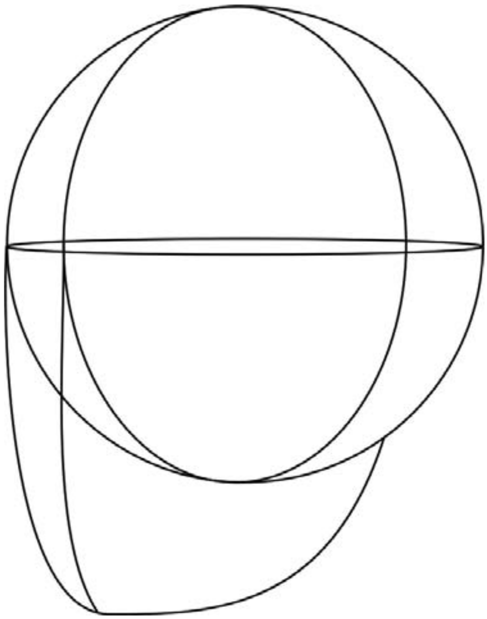


Figure 2.32
Draw the construction guide as a 3D shape.

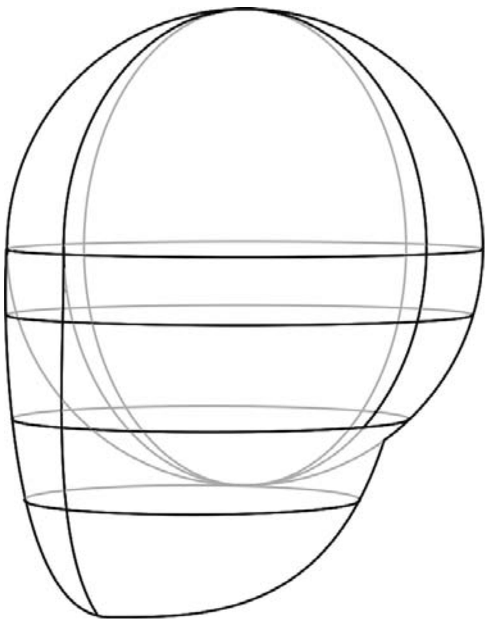


Figure 2.33
Finish drawing the construction guide in 3D space.

the time, you will need to create the construction guide as a 3D shape. Here's how:

1. Draw a circle.
2. Indicate the turn of the head to the left and the tilting of the head slightly down by drawing your cross lines as ovals, and draw the jaw line. (See Figure 2.32.)
3. Draw in the lines defining the eyes, nose, and mouth, as shown in Figure 2.33. The lines going around the back of the head are lightened so you can better see how they work.
4. The lines can now be used to define the location of the features. Figure 2.34 shows the construction guide over the face and the drawing once the guide is removed.

It takes a little practice to get the features in the right places using the construction guide. Remember that the construction guide defines the base of the nose where it protrudes from the face. The eyes, on the other hand generally recess in from the line. The top cross line is just about the level of the eyebrows. The sides of the face do not extend all the way to the edge of the circle.

Try drawing a few faces on your own. Figure 2.35 is a construction guide that you can copy and use to practice with.

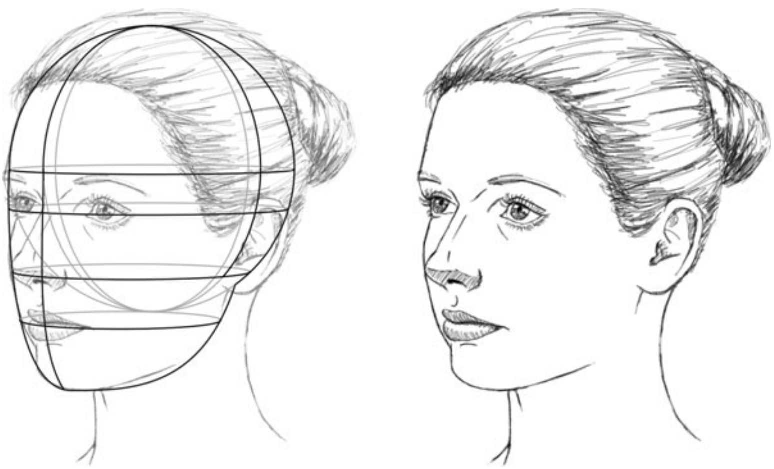


Figure 2.34
Use the construction guide to draw the face.

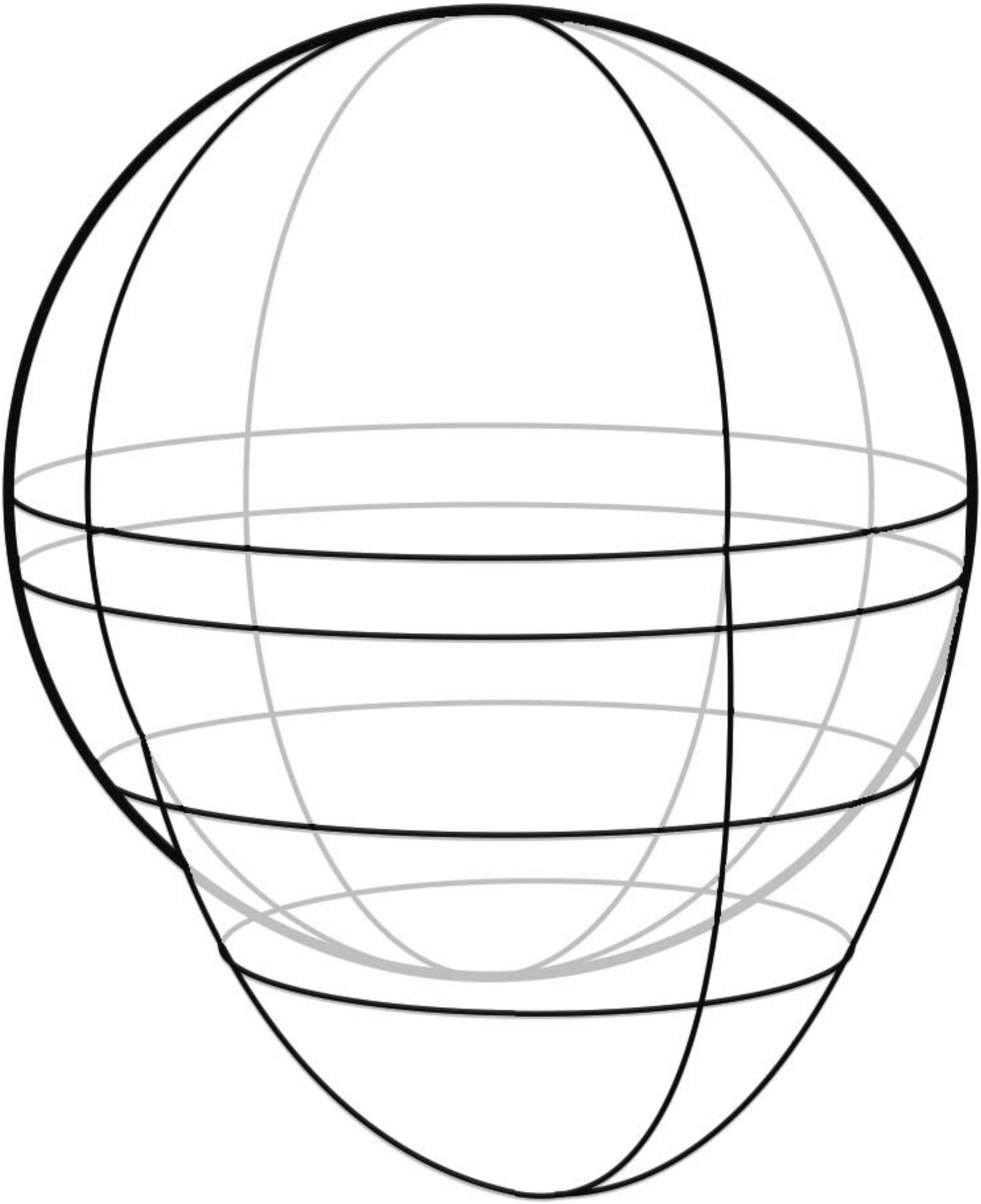


Figure 2.35
Use the construction guide to draw the head. (Photo-copy it first, so you can use it more than once.)

Facial Features

Drawing the head is easier if the artist is familiar with all the features of the face. Each feature is unique, and understanding them can improve your ability to draw good heads for your characters. A good way to understand how to draw the head is to isolate each feature and learn how to draw it. Try filling sketchbooks with eyes, noses, mouths and ears. Let's take a look at the individual features of the face and see how each one is drawn.

Eyes

The human eye is a spherical-shaped object recessed into the skull. You only see a part of it. It is covered by eyelids and protected by a ridge of bone that makes up the brow and cheekbones of the skull. Figure 2.36 shows the many parts of the eye.

Here are some tips for drawing eyes:

- Eyelids have thickness on the top and the bottom. It is most evident on the bottom lid.
- When drawing the lashes, plant the tip of the pencil at the base of each lash and release pressure on the pencil stroke as you draw in each lash.
- The highlights of the eye are direct reflections of the light source. It is always the brightest part of the eye—even brighter than the whites of the eye.

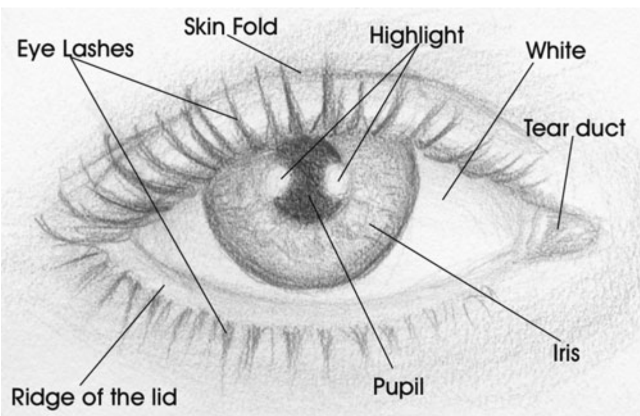


Figure 2.36
There are many parts that make up an eye.

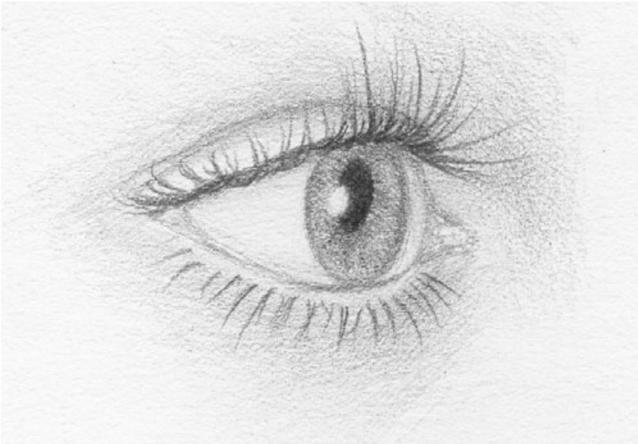


Figure 2.37
The eye bulges in the middle.

- There is often a fold above the eye that becomes more evident the more the person opens their eyes.
- There is a cast shadow below the upper lid on the eyeball.
- Remember to draw the tear duct on the inner side of the eye.
- The pupil sits behind the lens of the eye, not on the surface of the eyeball.
- The eye bulges from the skull at the center because of its ball shape (see Figure 2.37).

Nose

The nose is often one of the hardest features for a beginning artist to draw. That is because the nose is a protrusion from the face and is indicated mostly by delicate shading along the bridge. Often, the beginning artist will try to draw the nose using lines from the eyes. It is better to think of the nose as a protruding structure that blends in with the other structures of the face, as shown in Figure 2.38.

A nose is made up of bone, cartilage, and soft tissue. The bony bump often seen on the ridge of adult male noses is the transition between the skeletal bone of the nose and the cartilage, as shown in Figure 2.39. There are three pieces of cartilage in a nose: one along the bridge and two at the tip of the nose. If you feel the tip of

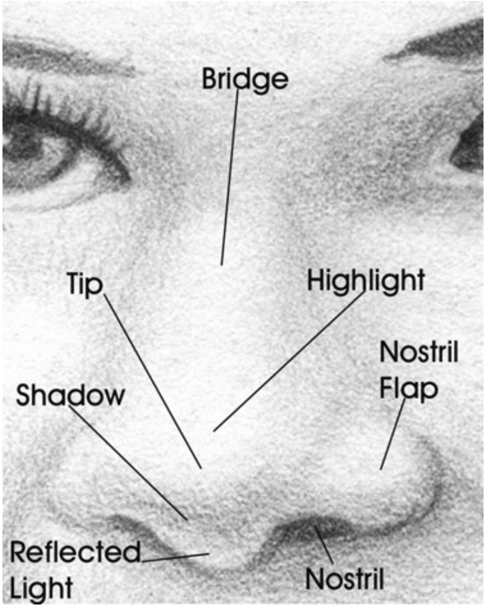


Figure 2.38
The nose is a protruding structure that blends in with the rest of the face.

Figure 2.39
Some noses have a small bump at the transition of the bone and cartilage.

your nose with your finger you should be able to sense the two plates and the small recess where they come together. With some noses, this recess between the cartilage plates can be seen.

Here are some tips for drawing noses:

- Both nostrils have a soft tissue flap that extends from the tip of the nose and around each nostril. The flap tucks in to the upper lip and often forms a slight outward bulge.
- The upper lip often reflects light to the lower part of the nose. The more ball-shaped the end of the nose is, the more this reflected light is evident.
- The highlight of the nose is usually above the tip of the nose.
- Noses are generally larger on older people because the nose continues to grow throughout our lives.

Mouth

The lips frame the mouth opening for the face, as shown in Figure 2.40. They are primarily fleshy tissue with underlying muscles that enable their movement. The lips cover the inner mouth and derive much of their shape from the teeth. If seen from one side or the other, there is a pronounced arch from the sides of the mouth to the middle as shown in Figure 2.41.

The lips form the most dominant feature of the mouth and are divided into two parts: the upper lip and lower lip. The upper lip is attached to the skull and the

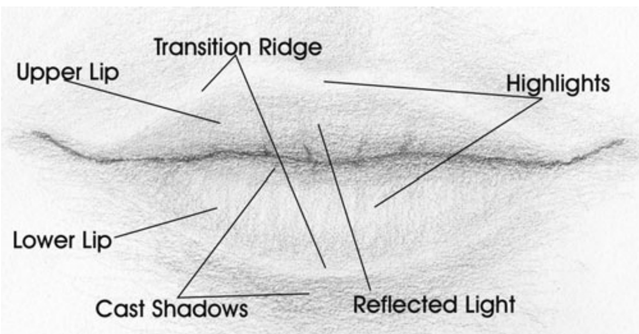


Figure 2.40
The lips frame the mouth.

Figure 2.41
The mouth curves around the face.

lower lip is attached to the jawbone. Because of the two separate bone attachments, the mouth has the widest range of change of any facial feature.

Here are some tips for drawing mouths:

- When drawing the mouth, the upper lip is usually in shadow from overhead lighting. The lower lip typically has more direct light and a highlight. The upper lip will often cast a shadow on the lower lip when the lips are together and on the teeth when they are apart.
- There is a ridge of transition between the skin of the face and the flesh of the lips. This area is slightly lighter than the surrounding skin and is more noticeable on people with darker flesh tones and on men.
- There is often a shadow beneath the lower lip, above the chin. The larger the lower lip, the more likely the shadow will be evident.
- When drawing the mouth open, draw the teeth as a single mass with shading. Come in later and define the individual teeth, as shown in Figure 2.42. Remember that the teeth are generally in shadow even though they are white.

Ear

Ears vary widely in shape and size from person to person. Some people have large ears and some have small. Some ears lay flat against the skull while others protrude out. Older people tend to have larger ears than younger people because the ear continues to grow throughout our lives. Even though the ear is not prominently placed on the front of our faces, it is still a significant facial feature and should not be ignored. Figure 2.43 shows the major parts of the ear. When

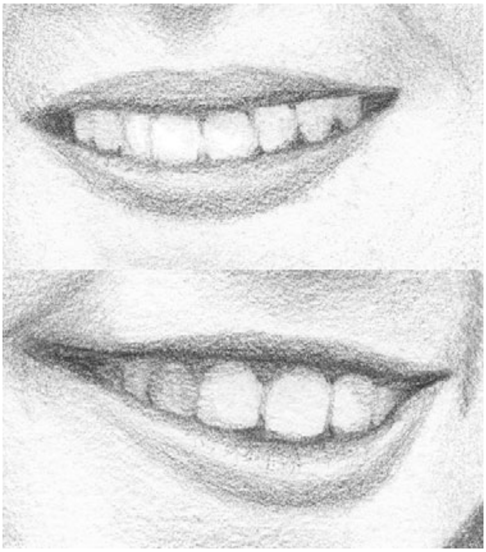


Figure 2.42
The upper lip usually casts a soft shadow on the teeth.

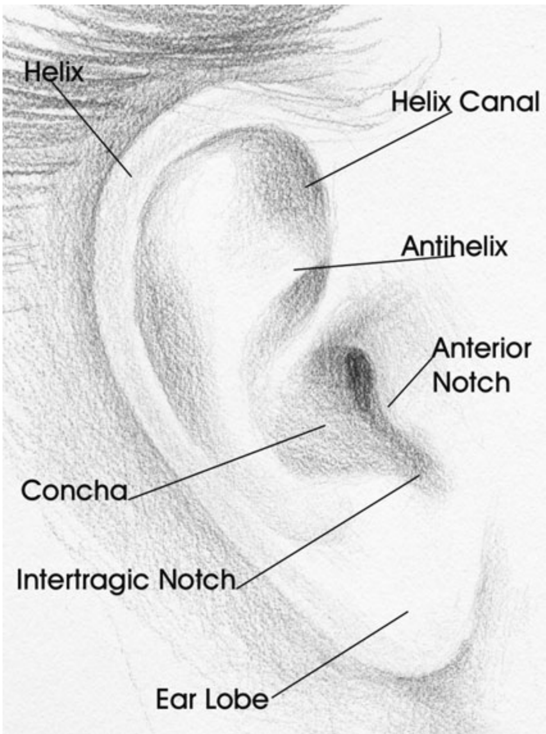


Figure 2.43
The human ear.

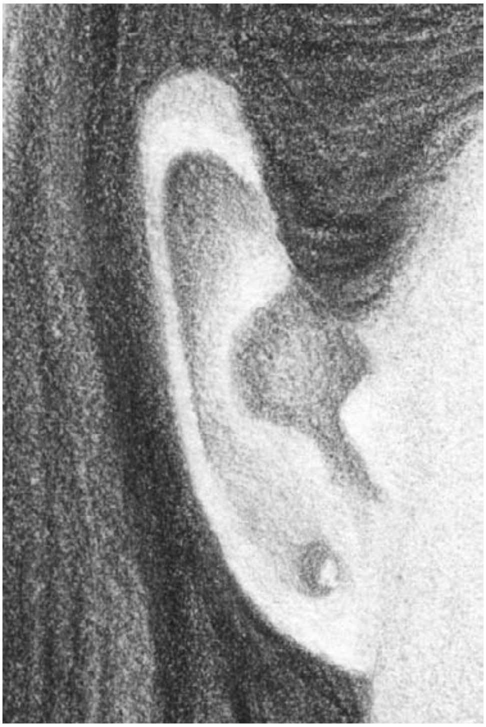


Figure 2.44
The ear is seen from the front in most portraits.

drawing the ear, think of the many shapes and shade them accordingly. There are many areas of cast shadow and reflected light. Most portraits view the ear from the front of the face, as shown in Figure 2.44.

The ear is basically a sound catching mechanism, and as such acts as a funnel for sound waves. The fleshy outer parts of the ear channel sound to the concha and then the inner ear through the ear canal. An anterior notch protects the ear cannel from damage. The ear is made up of a stiff yet flexible tissue that keeps its shape but can bend when needed. A ridge of skin called the helix surrounds the upper and back of the ear. The lower part of the ear is the forms a lobe and is the more flexible than the rest of the ear.

Hair

Okay, I know hair is not considered a facial feature, but it is part of the head, so I'll talk about it here. Hair is not an individual feature like a mouth or an eye, but rather is made up of many separate strands. Often, artists find hair difficult to draw because they don't understand how it catches light and how it flows. Figure 2.45 shows some of the basic aspects of drawing hair.

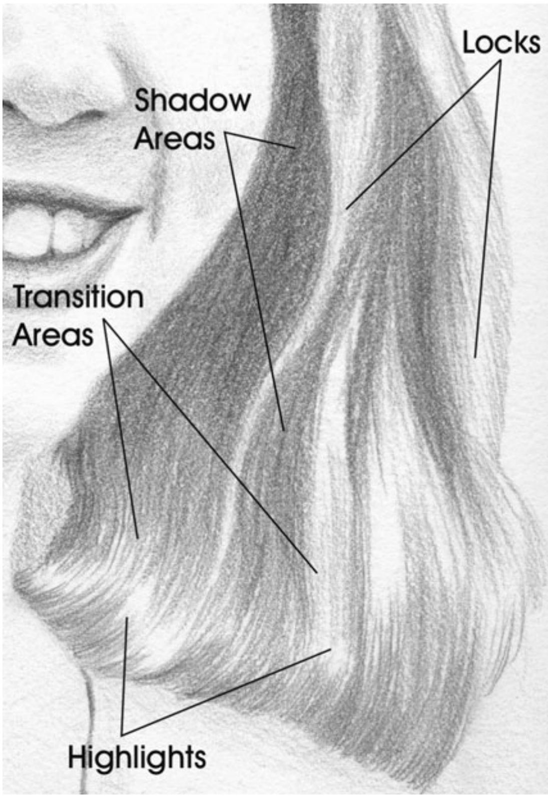


Figure 2.45
Hair is made up of many individual strands.

Here are some tips for drawing hair:

- Hair is usually shiny and catches light with many highlights. It is usually a good idea to look at the highlights and shadow areas of hair first. The highlights will be the white of the paper unless the hair is very dark. Draw in the shadow areas first. Then draw in the transitional areas with individual pencil strokes emanating from the shadow areas toward the highlight areas. Each stroke should start dark and then lighten as pressure is released from the pencil.
- Longer hair usually clumps in locks. A head of hair is made up of several overlapping locks. This is most evident in longer wavy or curly hair.
- When the hair is parted, as shown in Figure 2.46, there is usually a shadow area near the scalp and a highlight as the hair changes direction and lies against the skull.

Figure 2.46
A highlight runs parallel to a part in the hair.

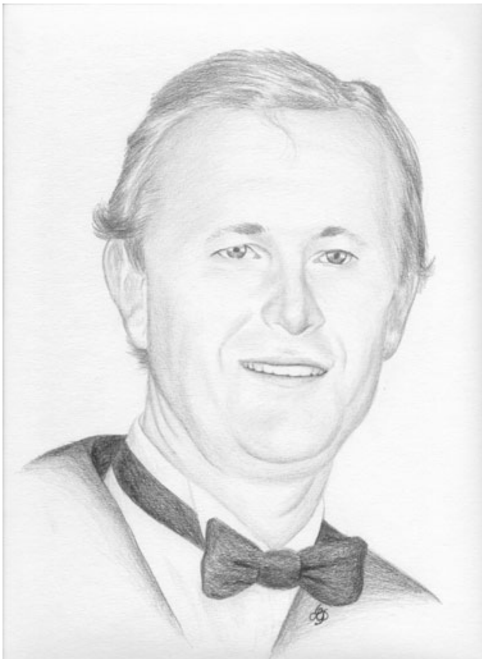


Figure 2.47
Put the features together to draw a portrait.

As you can see, there are many facets to each facial feature in a portrait. I suggest that you spend some time working on each feature until you feel comfortable drawing it. Fill some sketchbooks with pictures of eyes, noses, mouths, ears, and hair. When you feel you've mastered the individual features, try putting them together. Figure 2.47 shows a finished head.

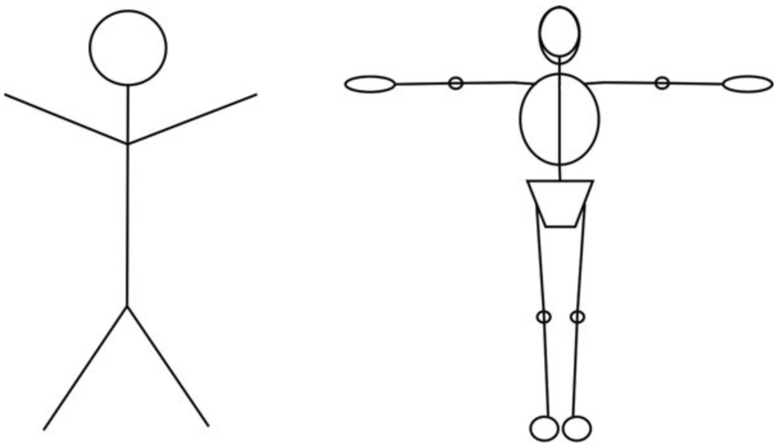


Figure 2.48
A stick figure is used as the foundation of character drawings.

Drawing the Full Figure

Drawing the figure is similar to drawing the head, except whereas the head is pretty much a ball with features added, the figure is a flexible form with extreme movement possibilities. Because the figure has such a dramatic range of motion, you need to interpret the dynamics of the range of motion within the character's pose. You also need to take into account things such as balance, distribution of weight, action, and proportions. This all sounds complex, but it can be simplified in a similar way to how you were drawing the head. If you can draw a stick figure, you can begin to draw characters. Figure 2.48 shows two stick figures side by side.

On the left is a typical stick figure. Next to him is the figure you will use as the foundation of your game characters. Figure 2.49 shows this stick figure from the back, front, and side.

The main difference for the new stick figure is that it includes an oval for the ribcage and a trapezoid for the hips. It is also proportionally correct for the average human character. In Figure 2.50, the stick figure is overlaid on a drawing of the ideal male proportions. Try drawing a few stick figures to get the proportions right.

Once you get the basics of the stick figure down, you can start moving him around to almost any pose, like the one shown in Figure 2.51. It is high stepping and appears to be moving in an animated way. Notice how much action you can achieve with just a simple stick figure.

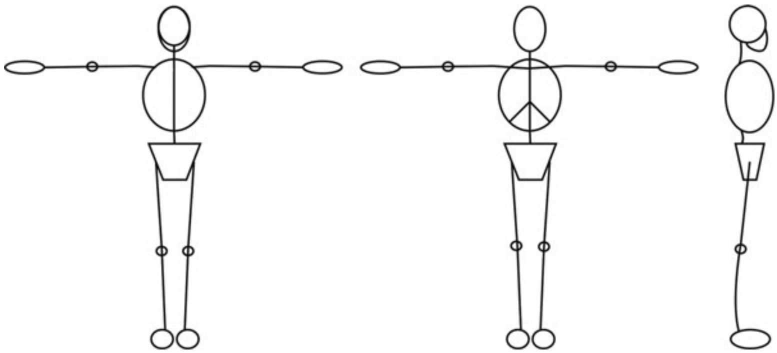


Figure 2.49
Draw the stick figure from the back, front and side.

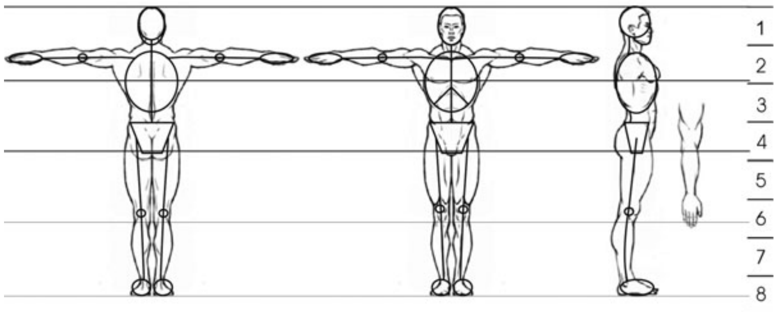


Figure 2.50
The stick figure is like the skeleton of the body.

Figure 2.51
The stick figure can be very animated.

Figure 2.52
The action line describes the action of the figure.

Creating the stick figure in action begins with a simple line called an *action line*. The action line is the main line of movement for the body. If you can, visualize the body moving in space; you can define a line of movement from the head along the spine and then through the legs. This line is the action line and it establishes the action characteristics of the figure. In Figure 2.52, you can see the beginning action line for a stick figure.

Over the top of the action line, the stick figure is loosely sketched in to indicate the position of the torso and the limbs, as shown in Figure 2.53. Notice how the figure follows the action line in this drawing.

In Figure 2.54, the drawing of the stick figure is complete. The figure looks like a superhero taking flight.

Notice that even though the character's right leg is lifted and pointing toward you, it can still be drawn with the stick figure. Limbs that are pointing toward the viewer are sometimes tricky for the beginner, however. The key is to draw the stick figure correctly. Look how that part of the character is drawn. To make the figure look correct, you have to think about where the leg is in space and what you would actually see from your point of view. Drawing the body correctly when part or all of it is rotated toward the viewer is called *foreshortening*. Figure 2.55 shows an arm with foreshortening. The arm is pointing toward the viewer.

Now you see the completed stick figure in action. This could be a superhero pose for a super-powerful game character. The stick figure works for any

Figure 2.53
Sketch the figure over the action line.

Figure 2.54
The finished figure looks like a superhero.

Figure 2.55
Use overlapping shapes to indicate a limb pointing toward you.

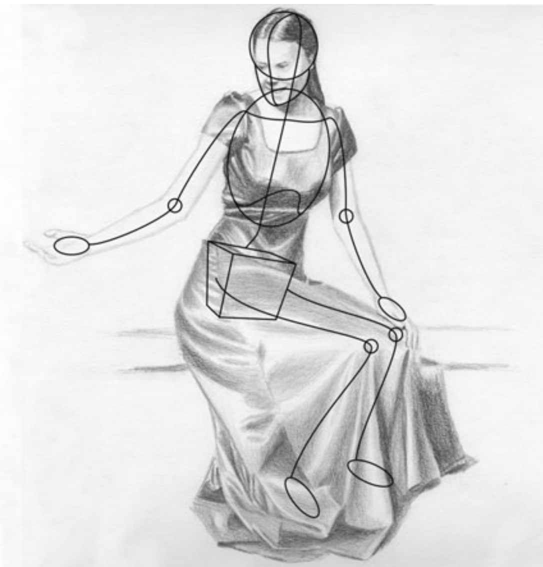


Figure 2.56
The stick figure works for any character drawing.

character you draw. Figure 2.56 shows a stick figure with the finished drawing super imposed.

Figure 2.57 shows the finished drawing without the stick figure.

Proportions of the Body

Body proportions vary a lot in human characters, depending on whether the person is short or tall, fat or skinny, heavy or light boned, and so on. Rather than

Figure 2.57
The character was developed by first using a stick figure for construction.

covering all of the differences, which would take more room than I have in this chapter, I will just cover basic proportions for an average game character of average height and build.

Figure 2.58 shows a drawing of a male character from the front and side. Note that an average male figure is roughly eight heads high, which is a little more than the average person in real life. We tend to be between seven and seven and a half heads high. The basic difference is that game characters tend to have slightly longer legs. This is purely a visual adjustment but it does make the character look better in the game. Some game characters exaggerate the differences. For example, some characters may only be five heads high while other character may be 10 heads high or more. Compare the two characters in Figure 2.59.

Note

> When drawing a figure, artists often use some readily available unit of measurement. Because the head is easy to distinguish and rarely changes shape, it is a common unit of measurement for drawing figures.

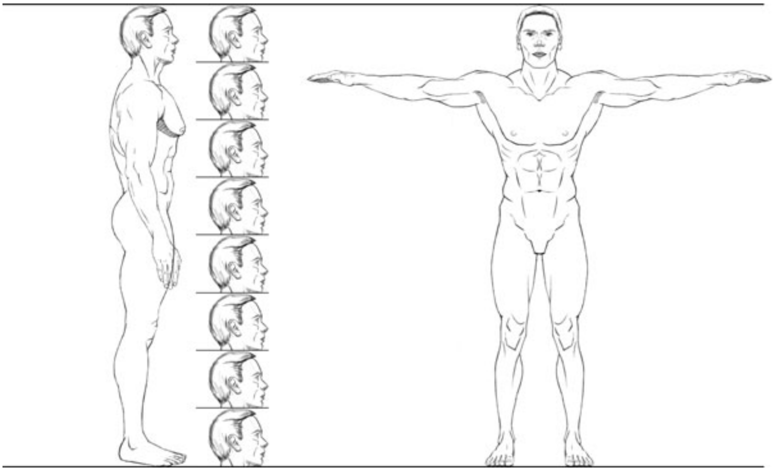


Figure 2.58
The male character is about eight heads high.

Figure 2.59
Many game characters are exaggerated.

Figure 2.60 shows a female character. Notice that she is also about eight heads high, even though she is most likely shorter than the male figure. This is because a person's head is usually proportional to his or her body.

There are a number of major differences between the male and female character. Male characters tend to be thicker and heavier in the arms and shoulders, while female characters are usually heavier in the hip area and upper legs. The male

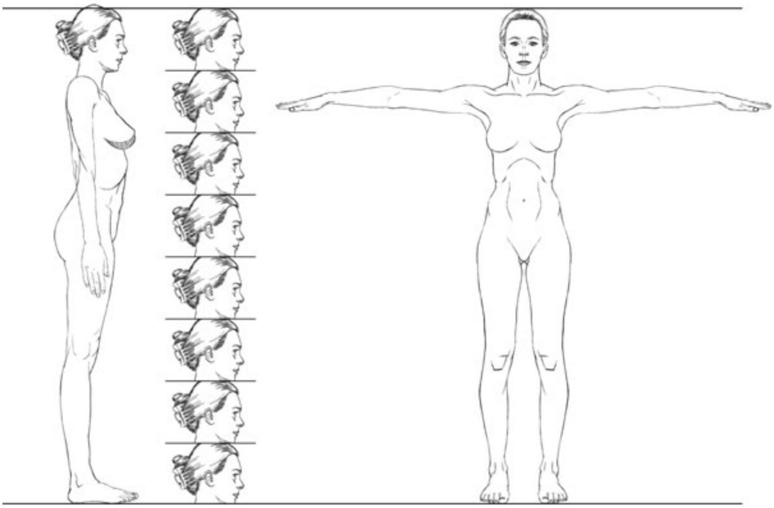


Figure 2.60
A female character is also about eight heads high.

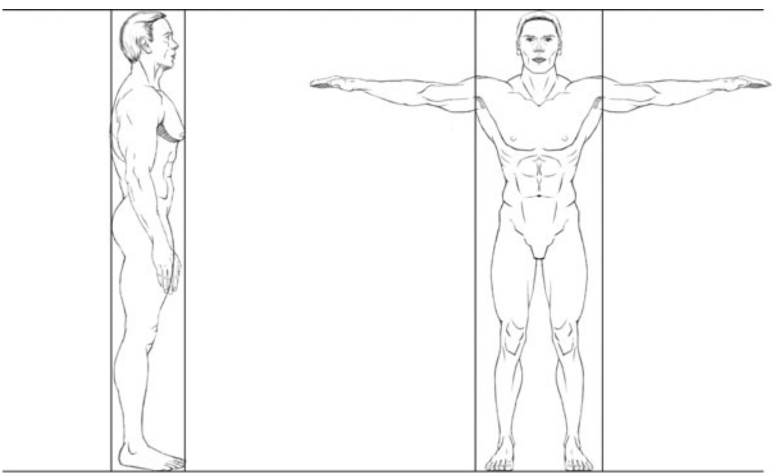


Figure 2.61
The male character is widest at the shoulders.

character is widest at the shoulders, as shown in Figure 2.61. In contrast, the female character is wider at the hips, as shown in Figure 2.62.

The male character tends to be thicker through the torso, particularly the upper torso and chest. The female chest is distinguishable by being narrower and has the addition of the female breasts. The male bone and muscle structure is heavier than the female's. Everything about the female character tends to be more delicate than the male. The bones are smaller and the features are more rounded on a female than on a male.

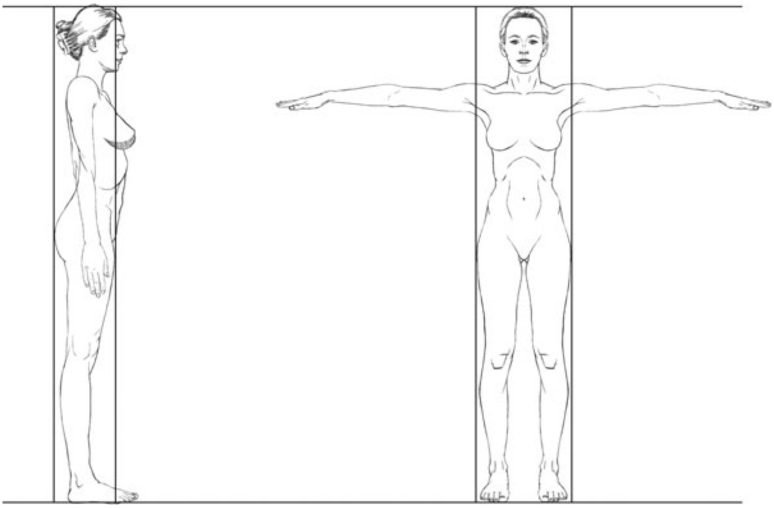


Figure 2.62
The female character is widest at the hips.

Beginning artists often miss a key balance point for drawing standing characters. When standing from the side, the body of both the male and female figure leans forward, as shown in Figures 2.63 and 2.64

Another interesting fact about both the male and female character is that the distance from finger tip to finger tip is equal to the height of the figure, as shown in Figures 2.65 and 2.66

These are only a few basic elements of a very complex biological system. To understand it fully, you will need to spend some serious time learning anatomy. When I was learning the figure, I spent a lot of time studying the individual bones and muscles to learn how they work in the body.

Hands

When drawing the head, you isolated the features to help better understand them and thus better understand how to draw the head. Hands deserve the same attention. They may not be a focal point like the eyes, but they are still extremely complex structures.

Note

The hand is a very flexible body part. The wrist itself is composed of eight separate bones to allow for extreme movement of the hand.

Figure 2.63
The center of balance for the female figure is about the front of the kneecap.

Most human characters have hands. Next to the face, the hands are probably the most expressive part of a character's anatomy. Hands are used for grasping, holding, pointing, fighting, touching, writing, squeezing, and so on. There is almost no limit to the number of things hands can do. Figure 2.67 shows a hand holding a gun.

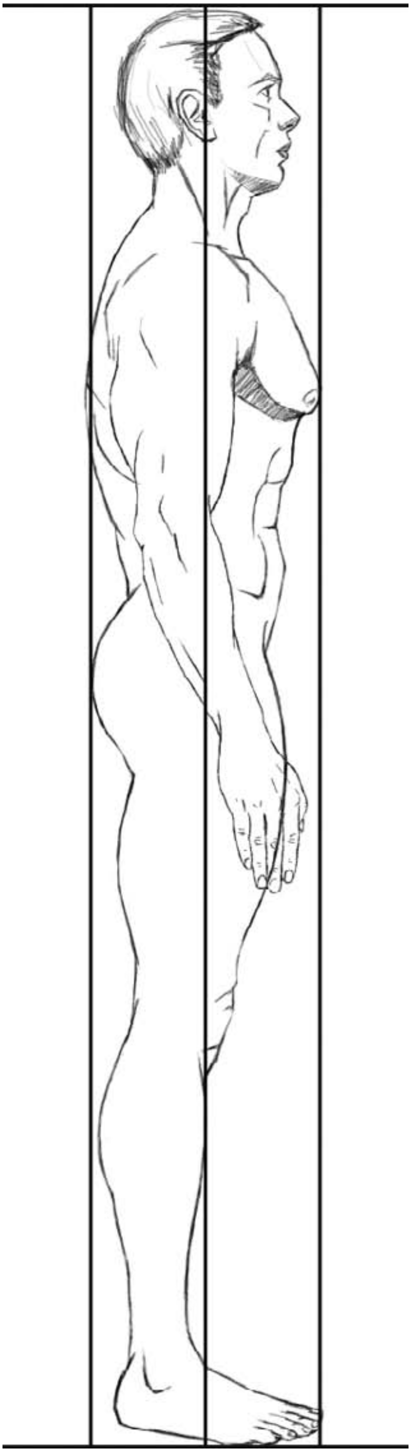


Figure 2.64
The feet are primarily behind the chest.

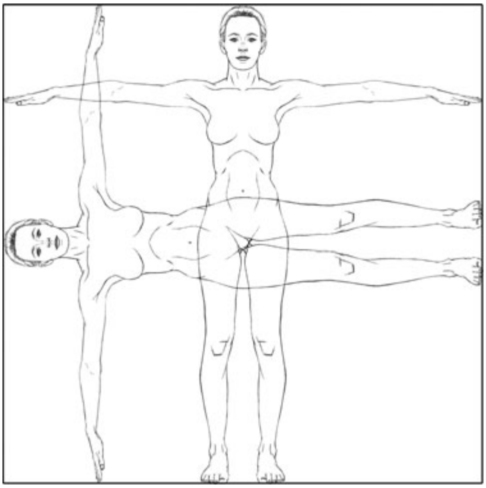


Figure 2.65
The arms are as long as the height of the figure.

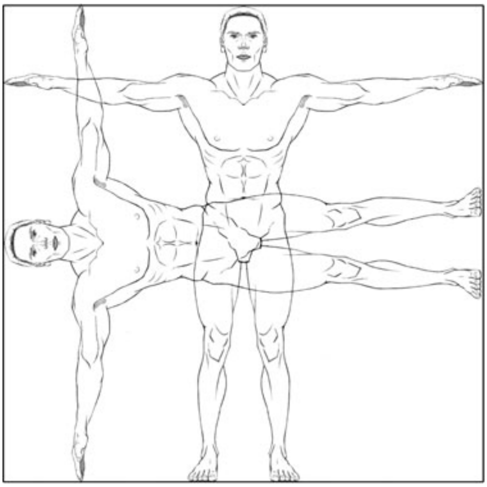


Figure 2.66
The figure's height equals its reach.

One of the best sources of practice for drawing hands is to draw your own hand. You can also try drawing your friends' hands. Figure 2.68 shows several hands drawn from different angles. Try drawing lots of hands until you start to feel comfortable with the subject. Think of the fingers and thumb as tubes connected by hinges. This should help you to better visualize the hand in three dimensions.

To better understand the hand, let's take a look at how the hand works. The hand has two sides: the palm, or grasping side of the hand, and the back, or non-grasping side of the hand. Figure 2.69 shows the palm of the hand.

Figure 2.67
Hands are used for a variety of purposes.

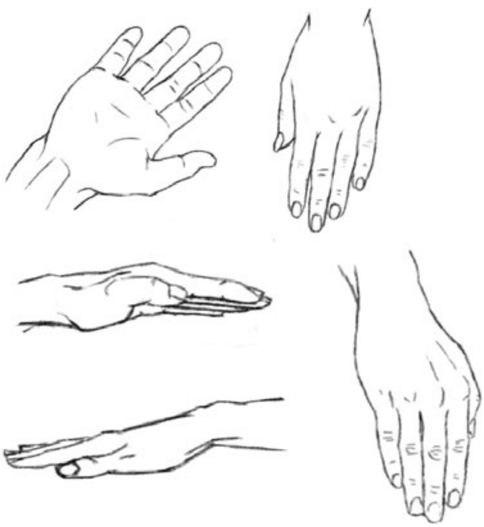


Figure 2.68
Draw hands from different angles.

The palm is characterized by being a concave surface surrounded by muscle tissue, the largest of which is the muscle that controls the movement of the thumb. The fingers are attached the end of the palm and the thumb is attached to the side. Tendons that extend through the palm to the muscles of the forearm control the movement of the fingers. If you look carefully at the underside of your wrist while making a fist you can see the movement of these tendons.

The back of the hand is shown in Figure 2.70. This part of the hand is characterized by knuckles, tendons, and veins, giving it a sometimes rough appearance.

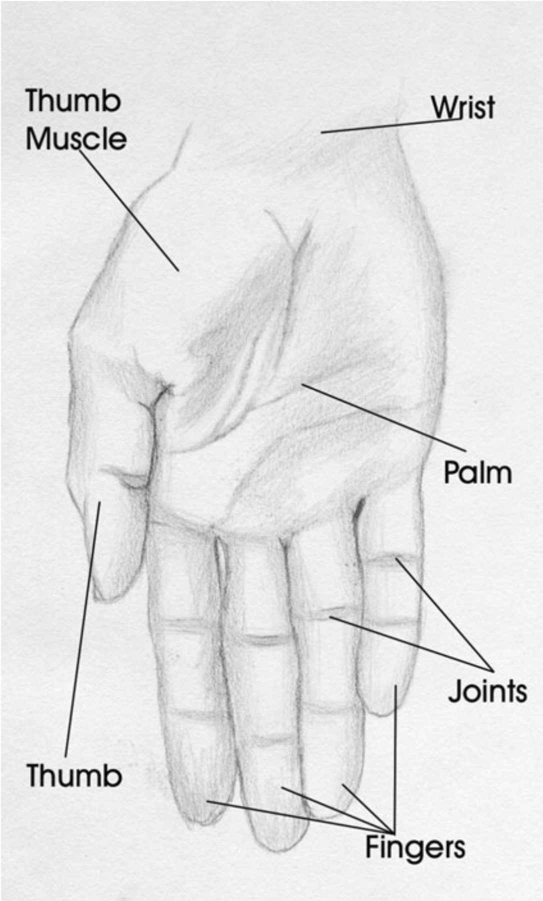


Figure 2.69
The palm is the grasping side of the hand.

The knuckles of the hand follow consecutive arcs that become more and more pronounced from the base of the hand outward, as shown in Figure 2.71. Notice that the second knuckle of the thumb is along the arc of the first knuckles of the fingers.

Putting It Together

I have covered many of the different aspects of drawing characters and even isolated some of the features. Now try drawing some characters of your own. Get some of your friends to pose for you so you can have something to look at when drawing. Figure 2.72 features one of my characters.

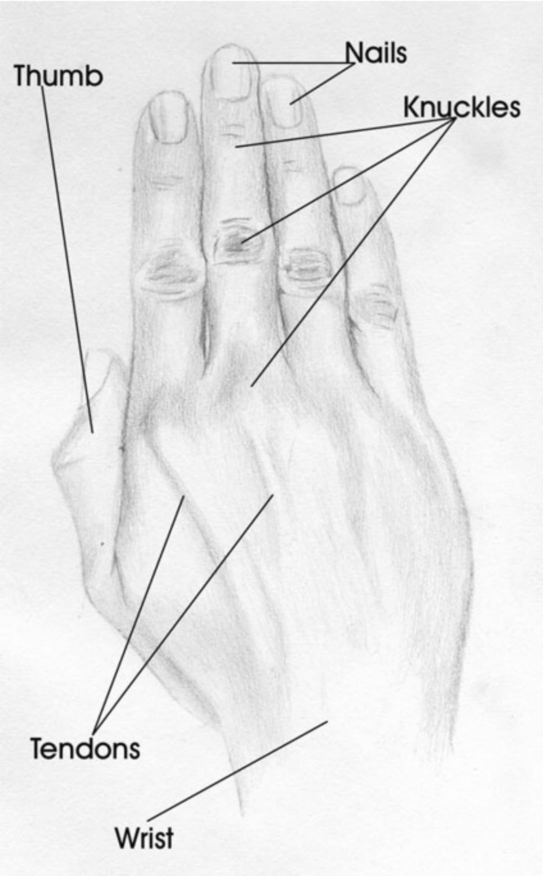


Figure 2.70
You can see the tendons on the back of the hand.

Character Exaggeration

Many games exaggerate the human character to achieve a desired effect. For example, an enemy may be given large shoulders and arms to make him more menacing, or a female character may have longer legs to make her seem more athletic and appealing. Exaggeration is a good tool for increasing creativity. Exaggeration is the process of emphasizing part of the drawing by making it larger or more distinct than it normally would appear. For example, the artist might enlarge the body of a character to make it more massive or give a character larger eyes to make her look cute, as shown in Figure 2.73. Figure 2.74 shows a sketch of a character whose arms and shoulders are exaggerated. The exaggeration

Figure 2.71
The knuckles of the hand follow arcs.

gives the character a more menacing look. Also, emphasizing the eyes and minimizing the rest of the facial features adds mystery to him.

One of the most common forms of exaggeration is giving the character an abnormally large head. There are usually two reasons for exaggerating the head. The first reason is that larger heads tend to make the character more childlike or cute. Thinking of the larger head being cute stems back to childhood. Baby's heads are quite a bit larger in proportion to their bodies than adults' heads are. The second reason for enlarging the head is that the head is the focal point of the character. In many earlier games, where character resolution was a problem, the head was enlarged to give the character more detail.

Exaggeration often works where other attempts at creativity don't. The magic of exaggeration is that it draws attention to the parts of a picture that the artist

Figure 2.72
Try drawing a few of your own characters.

wants to emphasize. That is why it works so well. All the artist needs to do is to decide what should be emphasized and then look at how to exaggerate it.

An assignment that might work well for an exaggeration approach is to create a final boss adversary for a fantasy role-playing game. The creature needs to be the most menacing and powerful creature in the entire game. It also needs to be intelligent and resourceful. The obvious solution for this problem is to create a dragon as the final boss, but that isn't very creative because many FRPs use

Figure 2.73
Exaggeration makes the character look cute and pleasant.

Figure 2.74
Exaggeration makes the character look mysterious and menacing.

dragons as their final boss creatures. Dragons, demon lords, arch mages—these are all common bosses for fantasy games. What could the artist do to create something that was more menacing yet original? Well, the answer might be in exaggerations.

Figure 2.75
Draw in the basic construction lines for the character.

Figure 2.76
Develop the character's features.

Figure 2.77
Finish the sketch by cleaning up the construction lines.

Drawing the Exaggerated Character

To get the hang of exaggeration, do the following:

1. Start a drawing by roughing in the basic shape of the character. Figure 2.75 shows the loose construction lines used to begin the sketch.
2. To enhance the feeling of a cute, innocent character, the character's eyes are enlarged and the character's clothing is oversized. The lines for the character are drawn in as curves, avoiding harsh angles. All these elements give the character a softer, more childlike appearance.
3. Continue to define the character's features, as shown in Figure 2.76.

Tip

When creating cute characters the artist should use simple forms and avoid excessive detail.

4. Draw in the remaining detail and clean up the sketch. Figure 2.77 shows the final sketch.

Summary

Character creation is one of the most rewarding and challenging jobs of the character artist. It is very rewarding for the artist to see a character that she or he designed in a game. It is also challenging to come up with characters that enhance the game design.

This chapter was devoted to character creation and drawing. You should now be familiar with many aspects of character creation. The topics discussed in this chapter include the following:

- Character descriptions
- Aspects of a character
- Writing a character description
- Thumbnail character sketches
- Drawing characters
- Drawing the head
- Features of the head
- Drawing the figure
- Drawing hands
- Exaggerating characters

This chapter dealt mostly with character sketches. Chapter 3 will take a deeper look into creating character illustrations and creating character templates.

CHAPTER 3

Character Illustrations and Templates

To create good character models, you have to start with good character designs that have ample detail to guide the artist in the modeling process. This detail may include a number of design elements like illustrations, model sheets or templates, photo references, texture samples, and so on. This chapter covers the process of creating character illustrations and then moves on to developing model sheets, which can be used as templates in Maya to build the model. This chapter is a continuation from the last chapter and will build upon some of the concepts covered in that chapter.

Character Illustrations

This section is about creating full-color character illustrations. It covers what character illustrations are and how they are used in games. It also gives some tips on creating good character illustrations. Two examples are given. The first example is of a simple character illustration and the second example is a more-detailed character illustration.

What Are Character Illustrations?

During the conceptual stage of a game's creation, what usually happens is that the character designer creates several sketches of a character, and then chooses the best one to become the actual character. At that point, the character illustration for the character is made. A *character illustration* is a full-color, detailed painting of a character by which a character designer communicates his or her vision of the character. The character illustration can be done in almost any color medium that the artist prefers, as long as there is sufficient detail to create the character for the game. Character illustrations do not have to be absolutely accurate, but they do need to convey the basic visual aspects of the character. For example, if the character is carrying any equipment or wearing specific clothing, this should be included in the character illustration. Emotional aspects of the character—for example, arrogance or bashfulness—can also be part of the character illustration.

Character illustrations are works of art that focus on a single character to define not only the look of the character, but his disposition and nature as well. When creating a character illustration, the artist must take into account the character's personality. If the character is a small, cute mouse, the illustration should convey that concept. On the other hand, if the character is a large, enemy alien cyborg, the illustration should look very different from the mouse illustration. Part of the character illustration's purpose is to help the modeler understand the type of character being created. If the character illustration can convey the message of the character's personality, the chance of the modeler's success increases dramatically.

Note

In contrast to the character illustration is the character template. A *character template* is a to-scale drawing of a character for use in building the character model. A character illustration is not used as a template for creating a 3D model; rather, it is used as reference for creating the model.

How Are Character Illustrations Used?

The primary purpose of a character illustration is to give information. The development team uses the character illustration for reference when creating the model sheets, 3D models, and animations for the character. That said, character illustrations have many other purposes:

- **Inspiration.** Developing characters for games is an interpretive process. Game characters move and react to events in the game. As technology has advanced, the complexity of the game character personality has expanded. The character illustration is a snapshot into the personality of the character. It should inspire the development team to create and animate the character.

- **Promotion.** Many character illustrations are used to promote the game. Often a character illustration is used in the advertising or promotional material because it is usually the first finished piece of artwork available.

- **Funding.** The character illustration is one of the first things a review committee might look at when determining the funding for a game. The characters in a game have a profound impact on a publisher's acceptance of the game for publication. Having very good character illustrations in the game design can be a plus for getting the game funded, while poor character illustrations can stop the funding very quickly.

What Makes a Good Character Illustration?

It takes many things to create a good character illustration. The first and most important aspect of a good character illustration is that the illustration should capture the look and personality of the character. There needs to be enough detail for the development team to see how to build the character and any clothing or accessories that the character might be wearing or carrying. The character illustration should also indicate how the clothing might move about the character as well as how the character itself might move. If the character is an old man with a cane who moves very slowly, the character illustration needs to show the old man bent over the cane, obviously having trouble moving.

Here are a few other things to consider when creating a character illustration:

- One of the basic components of a character illustration is that it is in color. Often, the character illustration is used as the color reference for the development team because character templates don't always have color, nor do character sketches. A good character illustration needs to have the correct colors for the character.

- Because character illustrations are often used for promotion or funding of the game, they need to be well designed and exciting to look at. The artist should take some time to plan the character illustration so it demonstrates good composition and has a strong visual impact.

- Ambiguity in a character illustration is usually not a good idea. If the illustration does not show the detail of the character clearly, it can cause misinterpretations when the character is built for the game. A good character illustration also should not feature harsh lighting that obscures parts of the character.

- A good character illustration should have correct anatomy. Many problems in building a character can be avoided if the character's proportions are drawn correctly. Most characters in games are 3D, so if a character is out of proportion or the limbs are not reasonable for movement in the game world, it can create problems for the development team.

Simple Character Illustrations

Not all character illustrations need to be finished to the same level of detail. If you know your character illustration will be used for promotion and to get funding, you'll want to pull out all the stops, making it as detailed as possible. If, however, your character illustration will be used only in the design document for the development team, the level of detail does not need to be as high. Likewise, if your character illustration depicts a character that is very important in the game, it will require greater detail than the character illustration for a minor character. Indeed, character illustrations for minor characters may not require much more than a simple color sketch. Here's a step-by-step outline of the development process:

1. The first step in creating a character illustration is to start with a good character sketch. The sketch can be drawn on paper or digitally, whichever you prefer. If you want to sketch by hand and then paint digitally, you will need to scan your drawing to make it a digital file. Figure 3.1 shows the base drawing for a muscular pirate from a high seas adventure game. Notice that there is greater detail in the upper part of the body, so the focus is on that area. Notice, too, that the character shows confidence and strength by his stance, with his legs braced apart and his arms at a ready position. The drawing is detailed but not shaded; the shading will be applied in the painting process.

Note

This example of a simple character illustration will show you how the majority of character illustrations are created. Later in the chapter, I'll give you a more elaborate example.

2. The next step is to develop the character sketch into something more detailed. Often, this development occurs digitally. That is, the illustration is created in a painting program like Corel Painter (used in this example) or Photoshop. Painter, in particular, is an excellent painting program for

Figure 3.1
Start with a drawing of the character.

character illustrations because it simulates natural media like oil painting and watercolor or pastels.

3. Painter's Airbrush tool is used to lay in a simple background for the illustration, as shown in Figure 3.2. The background colors are applied to the entire surface of the drawing. The background colors in the character will help unify the colors of the painting. The design principle here is to focus the attention toward the center of the painting. Indeed, the background is kept undefined so it does not take away any interest of the character.

Tip

When using the Airbrush tool, move from one area of the image to the next, painting in the colors. The areas that are not being painted should be masked off so you can apply paint only to the appropriate areas. Masking is automatic in Painter. Just select the area that you want to paint using one of the selection tools, such as the Lasso tool. All the areas outside your selection will be masked, meaning you can't paint over them. Lasso is the one most often used selection tools because it allows for full freedom of shape.

4. As shown in Figure 3.3, the next step is to paint in the initial flesh tones.

Figure 3.2
Add the background color to the painting.

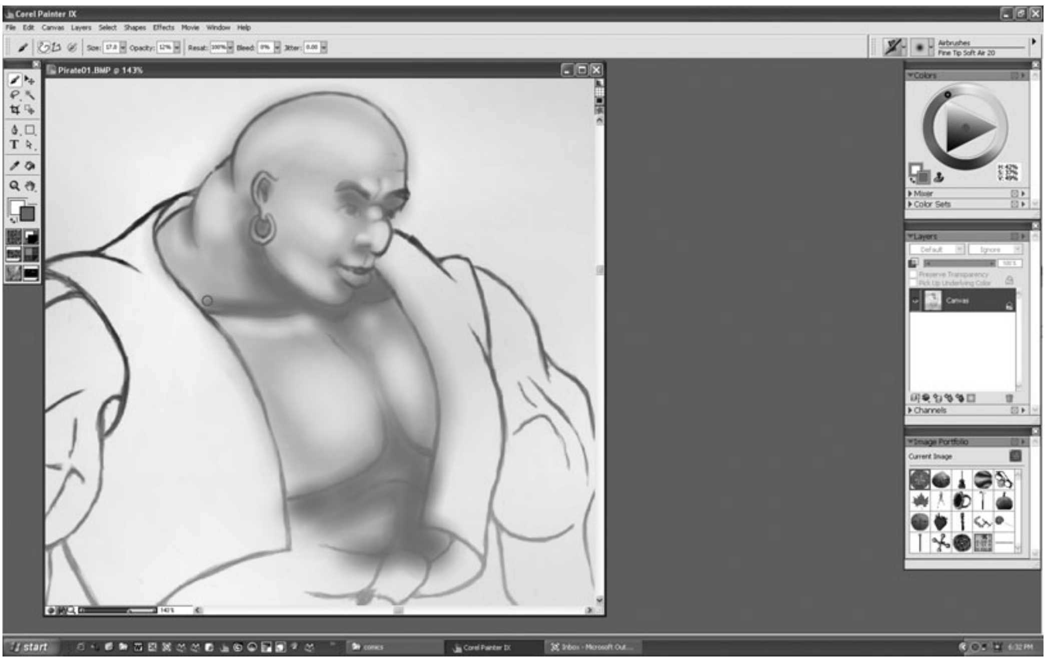


Figure 3.3
Paint the colors of the skin.

Figure 3.4
Continue to paint the flesh tones.

5. As the character illustration progresses, flesh tones are added to other parts of the character, such as the arms (see Figure 3.4).
6. This character is very muscular, so the bulging muscles are enhanced by the illustration. Figure 3.5 shows the flesh areas of the character painted.
7. The intense colors of the vest will require a mask to keep the reds from over spraying the areas already painted. I used the Lasso tool to define the area of the vest (see Figure 3.6).
8. I masked the remaining areas of the vest and the character's tattered shorts; Figure 3.7 shows the results.

Note

As you progress with a painting, try to keep the volumes simple yet at the same time give them dimension.

9. The airbrushed character is now painted, but he lacks the detail of the lines in the original sketch, as shown in Figure 3.8. The next step will renew that detail.

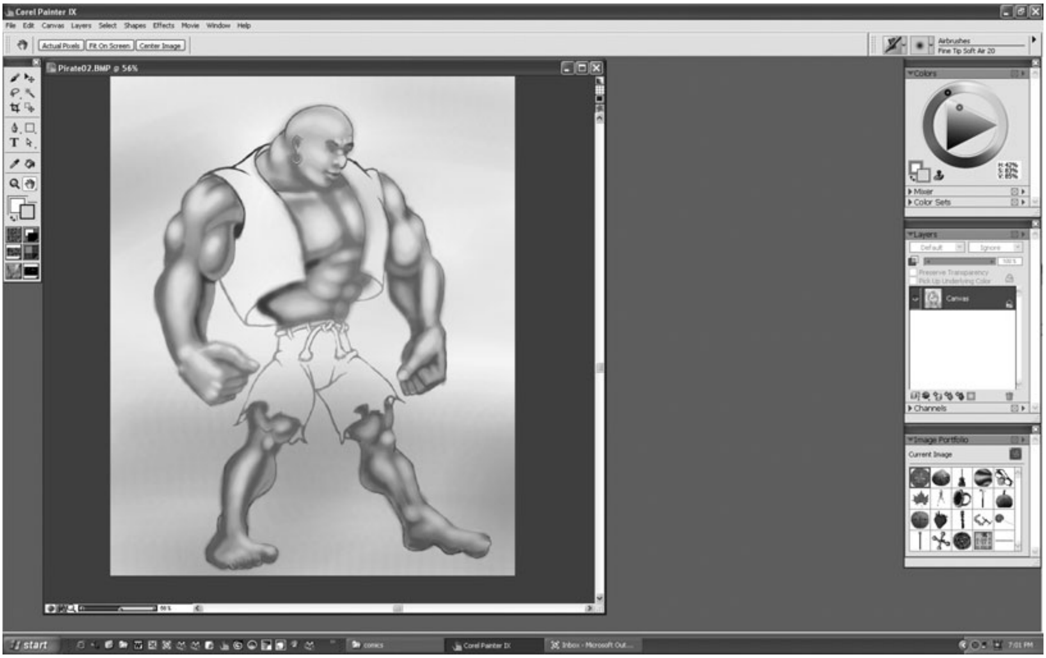


Figure 3.5
Give the character bulging muscles.

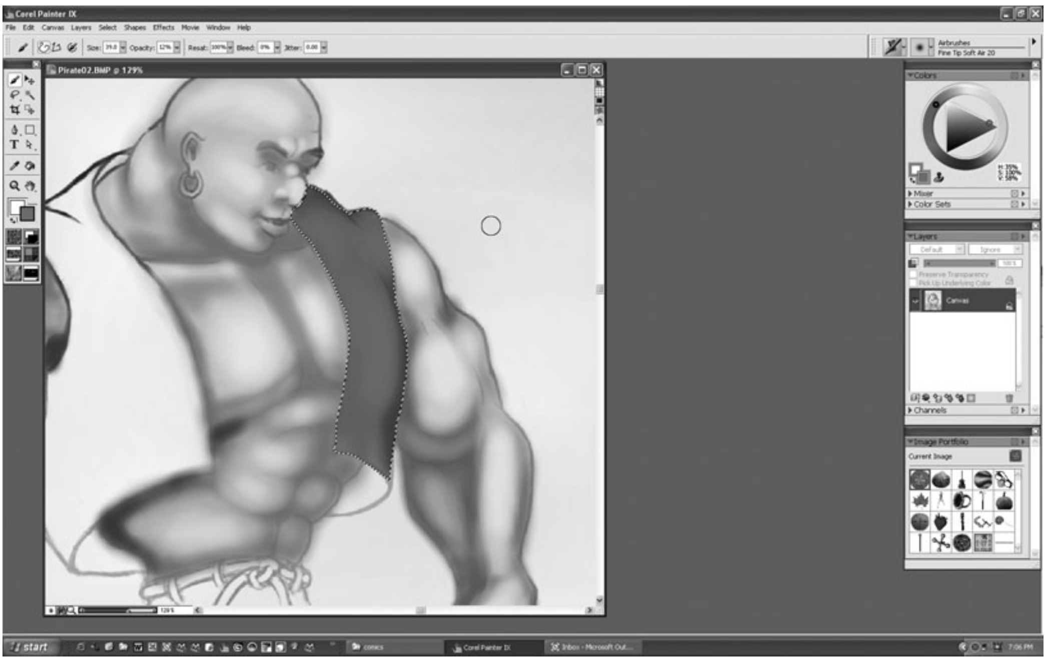


Figure 3.6
Select the area of the vest.

Figure 3.7
Continue to paint the character's clothing.

Figure 3.8
Bring up a saved version of the original line drawing.

10. The original drawing is copied and pasted it over the painting, with care taken to ensure that it is positioned exactly over the painting. The transparency of the drawing is then adjusted to 50% (see Figure 3.9).

11. The next step is to restore the vibrancy of the colors while retaining the lines of the drawing. This is done by dropping the drawing layer and adjusting the brightness down and the contrast up (see Figure 3.10).

12. The last step to finish the character illustration is to clean up any over-spray problems or any details like the pirate's ear ring and facial features. Figure 3.11 shows the finished character illustration.

Although this sketch of an exaggerated cartoon character features detail and dimension, it is not a full, finished character illustration. It is suitable for some game characters but not for the more important ones. With a little more planning and a lot more detail, the illustration could become a full, finished illustration. The next example shows you how a full, finished illustration is planned and completed.

Detailed Character Illustrations

A detailed, finished character illustration takes more planning and time to create than a simple character illustration. In many ways, the detailed, finished

Figure 3.9
Make the overlaid drawing 50% transparent.

Figure 3.10
Adjust brightness and contrast.

Figure 3.11
Finish the illustration by cleaning up.

character illustration is the closest thing to a finished work of art that a concept artist creates. When approaching the detailed character illustration, the artist should already have gone through the process of creating a character sketch. In fact, because of the time involved in creating a detailed illustration, the artist might have several sketches and thumbnails of the character. All this is to say that quite a bit of refinement of the character should occur *before* the work starts on the illustration. Here's a step-by-step outline of the development process:

1. The first task is to rough in the basic forms of the character. This character drawing is of a young woman in fine medieval clothing. She could be a noble woman or a young sorcerer in a fantasy game. Notice that line drawing does not have any shading; this is because its purpose is to guide the game artist through the process of painting the character. The shading will be added later during the painting process. Figure 3.12 shows the base line drawing, which was drawn digitally in Painter. Most character illustration is done digitally but some are created by hand using traditional media. Either approach is fine in most instances.

Figure 3.12
Create a detailed line drawing of the character.

2. Almost all detailed character illustrations include at least some background elements. For this picture, the background is a vine covered stone wall. Unlike the character, the background is shaded to show how the light and dark values will surround the character (see Figure 3.13).

3. The next step is to use a light wash of color to define the basic hues of the painting. Laying the colors in with washes helps to define the color scheme of the painting. Figure 3.14 shows the color wash painted over the top of the drawing. The underpainting is now ready.

4. When painting or drawing a character, I usually start with the face because that is often the center of attention. Figure 3.15 shows the beginning of the painting, starting with the eyes; Figure 3.16 shows a close up of the facial area of the painting. I work outward, painting in the features of the face, using the underpainting as a guide.

Figure 3.13
A background is added to the drawing.

Figure 3.14
Color is next added to the drawings.

5. I continue to work on the character's head, adding her hair and painting the flesh tones of the neck down to the collar of her tunic. Figure 3.17 shows the painting in progress with the head brought to a close-to-finished level.

Note

Some artists prefer to work the entire painting going from general to specific. I, however, like to work on an area of the painting, get it looking good, and then move on to the next area. I use this approach for two reasons: First, I usually paint in oils, and having wet paint over the entire surface just increases my chances of smearing portions of the painting. Second, establishing a near finished area of the painting helps me to gauge the rest of the picture against that area. It is kind of like getting one area to look good and then spreading that out over the rest of the painting.

6. The next area that I work on is the character's upper torso down to her belt. Her tunic, which is basically white with gold trim around the shoulders and

Figure 3.15
The painting begins with the character's head.

Figure 3.16
The eyes are the first features to be painted.

Figure 3.17
The head is brought to a near-finished state.

borders, needs to drape across the form of the character's body correctly in order to show the underlying structure of her upper torso. Figure 3.18 shows the picture in progress with the upper torso painted. Figure 3.19 shows a close-up of how the gold trim was rendered—by painting in the basic folds of the cloth and then working in the detail of the design.

7. Next, I work on the character's arm and her belt. Figure 3.20 shows the progress.
8. The belt is painted in much the same way as the gold trim of the tunic, as shown in the close up in Figure 3.21. The belt also has a small gemstone in the buckle.
9. The character's hand, though small compared to the rest of the painting, is very intricate. Figure 3.22 shows a close-up of the hand. Notice the many subtle shadows around the palm and fingers.

Figure 3.18
The upper torso is painted next.

Figure 3.19
The detail of the design on the trim is added after the basic folds are painted.

Figure 3.20
Develop the painting around the character's belt

Figure 3.21
Painting the belt is similar to painting the trim.

10. When the arms and the belt are painted, I paint the rest of the front of the character's tunic and the small decoration extending down from the belt. Because the character is walking, the tunic is draped over the top of her leg. Figure 3.23 shows the painting.

Figure 3.22
The hand is a complex form and requires delicate shading.

Figure 3.23
The tunic follows the form of the character's leg.

Figure 3.24
The character's pants and boots are painted next.

11. I move down the character's body to her legs, painting her pants and boots as shown in Figure 3.24. Figure 3.25 shows the characters hip and upper-leg area. Notice how the clothing indicates the form of the character's body. While many of the folds in the clothing are simplified, they still give the impression that there is a three-dimensional form underneath.
12. I move to the background. Because the background exists solely to enhance the character, I use subdued colors to paint in the rocks of the wall behind her. I also work with a textured surface, as shown in Figure 3.26.
13. I move to the cobblestones on the street. This area is darker than the rest of the picture because I want the attention focused on the area around the character's head. By decreasing the value differences and reducing the amount of detail here, I indicate to the viewer that this is a less important area in the illustration. Not every area of an illustration should have the same level of detail and finish. Figure 3.27 shows the results.

Figure 3.25
Folds in clothing need to follow the underlying form.

Figure 3.26
The Rocks are painted next.

Figure 3.27
The bottom of the picture is not as detailed as the top.

14. The last part of the picture that I paint is the vines that cover the wall, as shown in Figure 3.28. The vines are strategically placed to accent the most important part of the character, her head. Notice how they frame her head and shoulders, giving that area the greatest contrast from light to dark. Even though the tunic is white, it is set next to a mid-tone wall where the character's head is next to an almost black background. Figure 3.29 shows a close-up of the character's head.

This is just one example of a more detailed character illustration. The detail in the illustration will help the development team to define the overall character better than if there was only a simple character illustration. The detailed illustration also can be used to help promote the game.

Figures 3.30–3.32 are other examples of detailed character illustrations.

Figure 3.28
Paint the vines in next.

Figure 3.29
The dark vines in the background provide contrast for the character's head.

Figure 3.30
Yes, chickens and knights can go together.

Figure 3.31
The guard is accompanied by his trusted marsupial.

Figure 3.32
Using monochromatic colors everywhere but the character's head brings the viewer's attention to the area of the character's face.

Character Model Sheets

Character illustrations are great for all the reasons mentioned so far in this chapter, but they alone are not enough to give the 3D modeler enough information to accurately build a model. Another critical piece of artwork for the concept artist to create is the model sheet, sometimes called the character template. This section is about creating model sheets. It covers what model sheets are and how they are used in game development. It also covers creating base templates for male and female characters.

What Are Model Sheets?

A *model sheet* is a template created by the concept artist to help the development team create accurate models of the characters. The model sheet is an orthographic drawing or painting of the character. *Orthographic drawings* are drawings that show the character from flat perpendicular points of view. Each point of view is designed to give the development team vital information about the character. Figure 3.33 shows a model sheet for a boy ninja character.

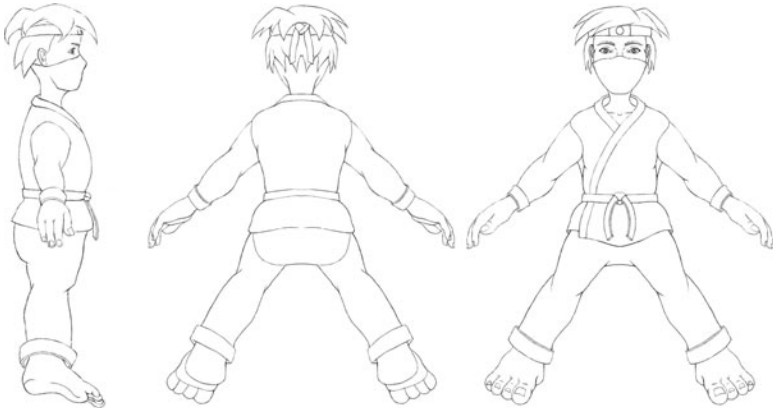


Figure 3.33
The character is seen from the side, back and front.

Many characters in games today are highly detailed. Games are getting closer and closer to reality in their graphics. Things that were impossible only a few years ago, such as cloth movement, facial animation, specular lighting, bump mapping, and transparency, are commonplace today. The movement to realism in characters is putting a lot of pressure on concept artists and development teams to create better, more-lifelike characters. The model sheet is the vital information link between the concept artist and the development team.

Model sheets are a form of art borrowed from the motion picture–animation industry. When animators started drawing characters, they had to keep the character looking consistent in all the scenes. Back then, all the frames of animation were drawn by hand. The problem became even more daunting when multiple artists worked on the same character. To help solve the problem, the industry came up with a system to help the animators keep their drawings consistent. The model sheet was a big part of that system.

Another industry where model sheets are used extensively is the comic-book industry. Like motion-picture animation, comic-book animation has several artists working on the same character. Although the requirements for consistency are not as rigorous in comics as they are in motion pictures, they are still fairly rigid.

Unlike motion pictures and comics, games do not have as much hand-drawn art. Instead, games primarily use 3D models. The advantage of using a 3D model is that the character is consistent because the same model is used throughout the game. However, the need to keep the game character consistent with the designed character is still important. The model sheet helps address that need.

In many ways, model sheets are similar to drafting. The character is seen from different angles in isometric views. *Isometric* means without perspective. So in other words, the view is flat to the viewer, with no distortion for distance. Isometric drawings are more accurate than perspective drawings because the elements can be plotted directly from view to view. To see how this is done, do the following:

1. Start with a simple drawing of the head, as shown in Figure 3.34.
2. Next, draw the side view of the head to the right of the current drawing. Use lines from the original drawing to line up the features so both drawings are accurate. See Figure 3.35 for an example.

Note

It is especially important to be accurate when you are drawing model sheets because the drawings are used as templates to create 3D models.

3. Next, create a side view above the side view you just drew, but rotate this new one 90 degrees counterclockwise. Finally, draw the top of the head. (See Figure 3.36 for an example.)

The finished drawing is an orthogonal view of the head. *Orthogonal* means that the subject of the drawing is seen from multiple angles. Each angle shows a different view of the character, and the combination should provide enough information for a model builder to create the character.

Figure 3.34
Start with a drawing of a head.

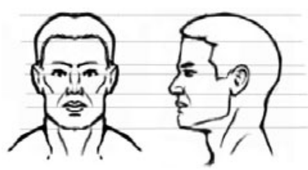


Figure 3.35
Project the side view of the head to the right of the original drawing.

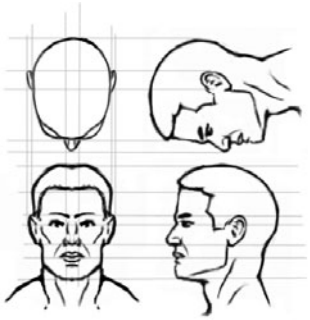


Figure 3.36
Draw a rotated side view and the top of the head.

How Are Model Sheets Used?

The problem with using character illustrations for reference when creating 3D models is that the illustration only shows the character from one direction, typically the front. If all you had was the character illustration, you would have no idea of what the back of the character looked like. To accurately create a game character, the development team needs to know what the character looks like from all angles. Model sheets are used for just this purpose.

Some model sheets are in color. Figure 3.37 shows a model sheet of an Earth elemental with color added to it to show how his skin will be different from a normal character.

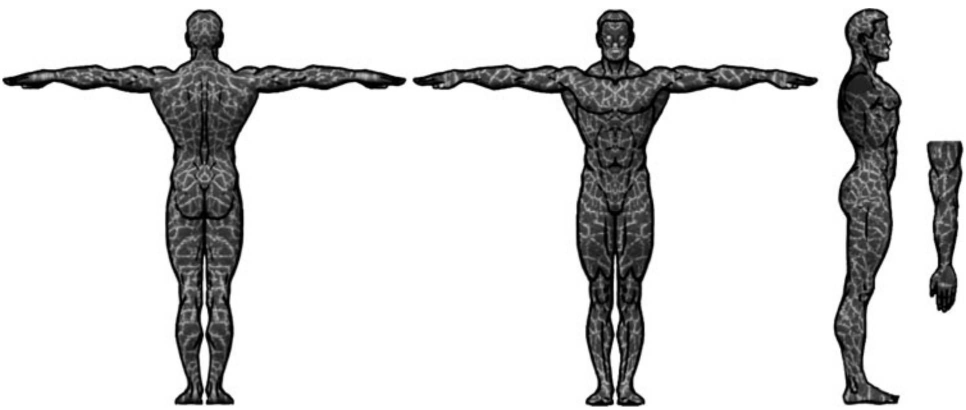


Figure 3.37
Some model sheets are in color.

Most model sheets, however, are black-and-white line drawings because they tend to work better as templates when the team is developing the character. A *template* is a guide used to create 3D models. Figure 3.38 shows a model sheet used as a template in a 3D program. Figure 3.39 shows a 3D model of the torso as it is being built, using model sheets as a guide in a 3D modeling application.

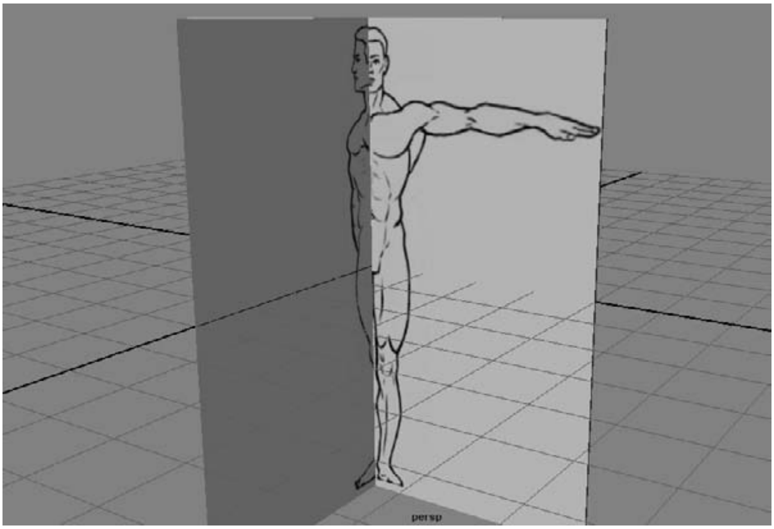


Figure 3.38
Model sheets are used as templates to create 3D models.

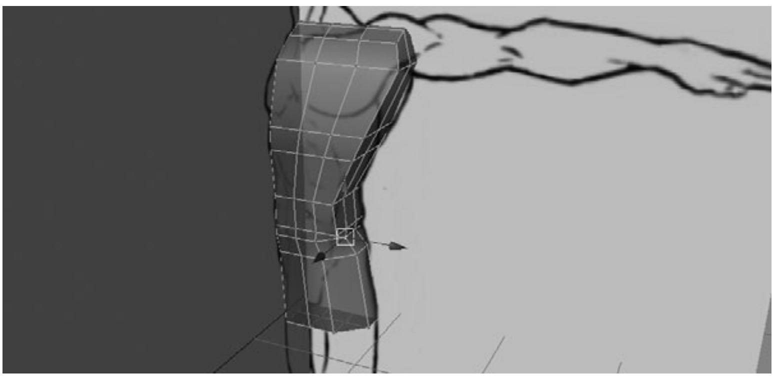


Figure 3.39
The torso is being built with the use of model sheets.

Note

Notice that the model sheets intersect each other in the 3D program. This is why all drawings need to be accurate and to scale for the template to work. When the 3D artist sets up the model sheets to create a model, some very undesirable results can occur in the modeling process if the drawings are not accurate.

It is helpful to understand how the template will be used in the 3D modeling program so that you will know why it needs to be so accurate. If the purpose of the drawing is for reference only, accuracy is not as critical as it is when the drawing is used directly in the development of the game art.

In addition to helping the development team create the characters for the game, model sheets are also good because they force the concept artist to fully design characters. Character illustrations do not contain detail from all angles and, therefore, don't fully design characters. Because creating a model sheet requires the artist to draw the character from several different perspectives, it forces the artist to fully design the character. Character designs are typically not complete without the model sheets.

Creating Base Model Sheets

Human characters are the most common type of character found in video games. A good place to start when you are creating model sheets is to develop base human characters with the proportions most commonly used in games. You can later use these base characters to create other characters without having to start from a plain white sheet of paper every time. These base characters are similar to mannequins in a clothing store; all the artist needs to do is dress them in different clothing to create different characters.

The Male Character

The human male character is very common in games. He is usually a powerfully built character with well-defined muscles. He can easily carry a 200-pound gun in one hand. Figure 3.40 shows the front view of a base male character.

Notice how the character in Figure 3.40 is drawn with the arms extended. That's because when an artist skins a 3D model (in other words, attaches the model to a set of bones for animation), the arms need to be out away from the body so that when the model is built, it will attach to a skeletal system correctly. That is why you see most 3D models with their arms straight out to the sides. Some 3D artists

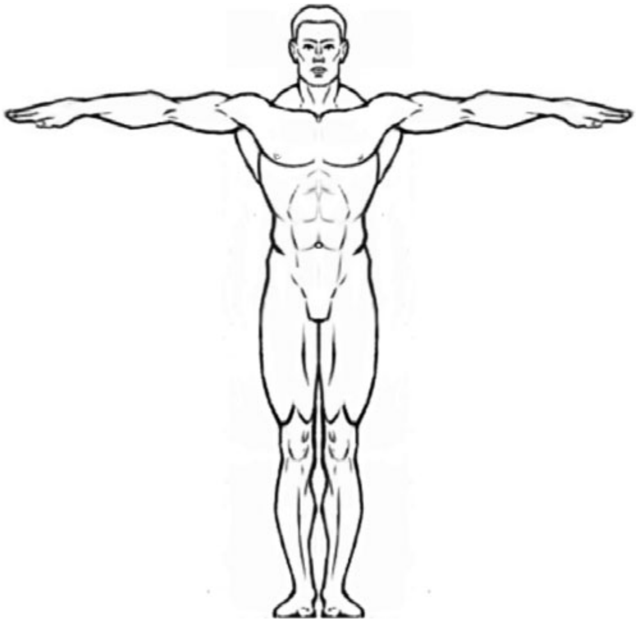


Figure 3.40
Draw the character from the front.

prefer to have the arms at a 45-degree angle because of the distortions in the shoulders when the arms are straight out to the side. The disadvantage to the 45 degree angle is that if the character raises his arms directly up in the game, the shoulders may have problems on a polygonal model.

A simple way to create the back view of the model sheet is to use a piece of tracing paper. Because the drawings are orthogonal and isometric, the front and back views will have exactly the same outside proportions. Lay the tracing paper over the drawing of the front view and trace the outline of the figure. Then, draw in the surface detail for the back. Figure 3.41 shows the back view laid over the front view.

The side view is a little trickier. To render it, return to the front view and extend guidelines from the major features to the side. Then use these guidelines to draw the side view, as shown in Figure 3.42.

The development team can usually get all the information it needs in order to create the figure from the back, front, and side views, with the exception of the arms. The arms need a top and sometimes a bottom view to help show depth, similar to how the body needs a side view. Draw the top view of the arm by extending lines from the front view upward, as shown in Figure 3.43.

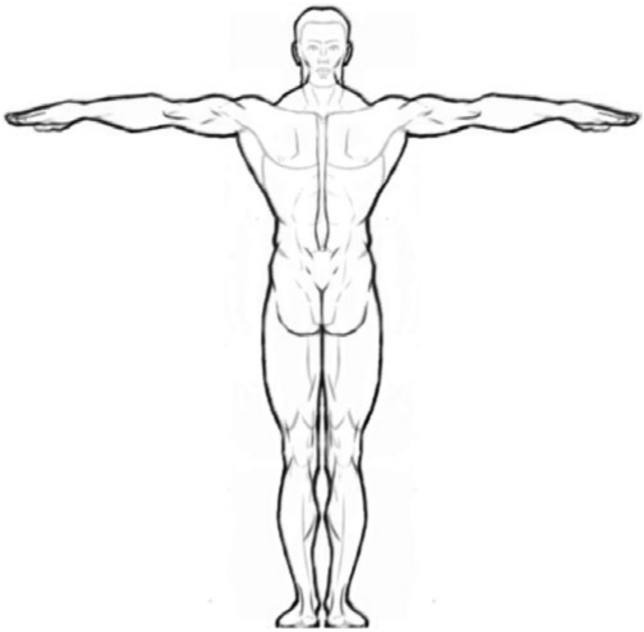


Figure 3.41
Lay the back view over the front view.

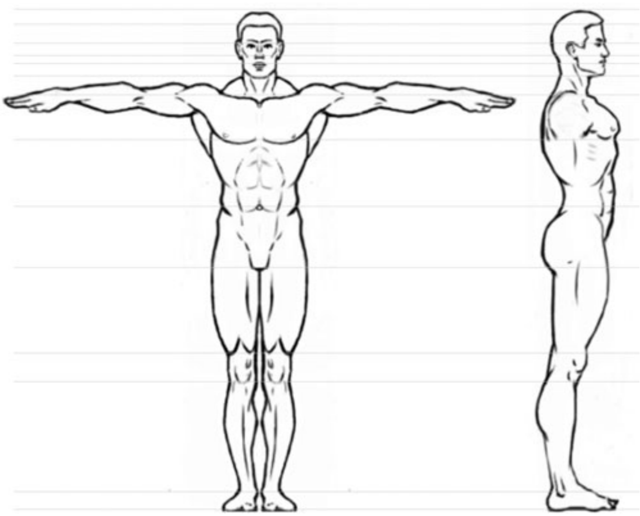


Figure 3.42
Use the guidelines to draw the side view.

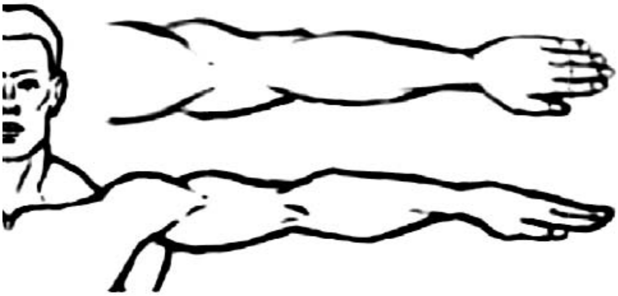


Figure 3.43
Extend guidelines upward to draw the top view of the arm.

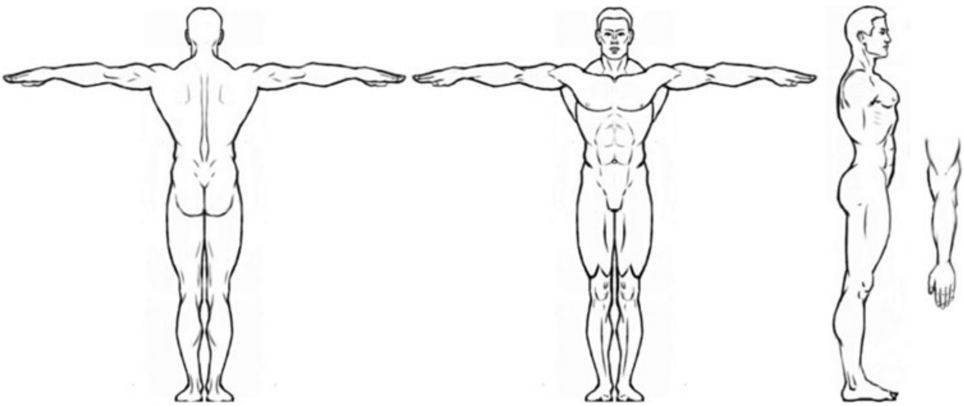


Figure 3.44
Each drawing should be proportional for the base male character.

Now the base model sheets for the male character should be complete. Figure 3.44 shows the completed model sheet of the base male character.

The Female Character

Female game characters generally are more slender than male characters. Like the male characters, they are idealized in most cases. In fact, some are impossibly idealized. Even though in games a female character is powerful, she is designed to also be beautiful, so typically she does not have the same bulging muscles that a male character has—even though female characters tend to be just as strong as male characters. Like the male character, a female character can carry a 200-pound gun easily in one hand.

The base female character is developed the same way as the base male character. Start with the front view. Figure 3.45 shows a front view of a base female character.

Notice that in addition to being more slender than the male character, the female character is also drawn with her feet pointed slightly downward. This is because most female characters in games wear heels. By superimposing the female character over the male character, the differences between the two become very evident (see Figure 3.46).

The female character model sheet is developed using the same steps as used for the male character. Figure 3.47 shows the completed female character base model sheet.

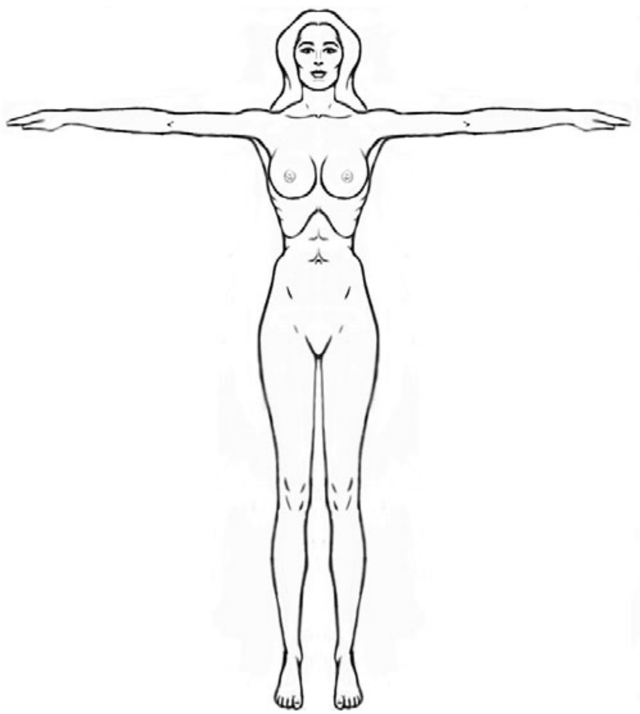


Figure 3.45
Start the model sheet by drawing a front view.

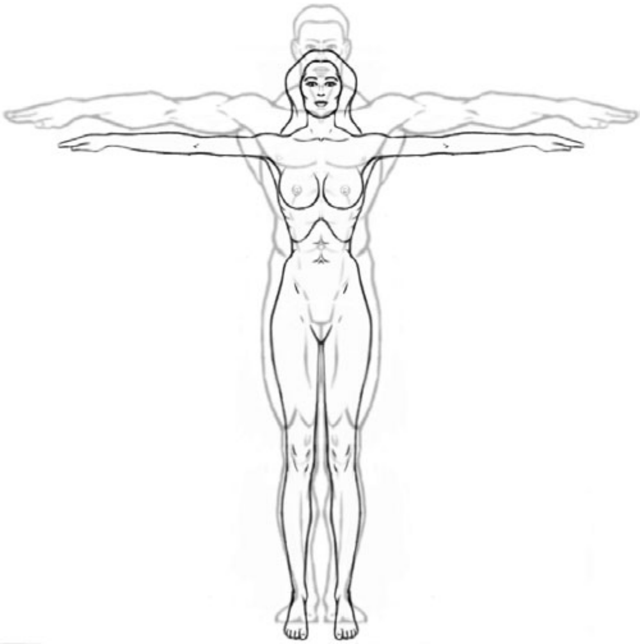


Figure 3.46
When the female character is superimposed over the male character, the differences become evident.

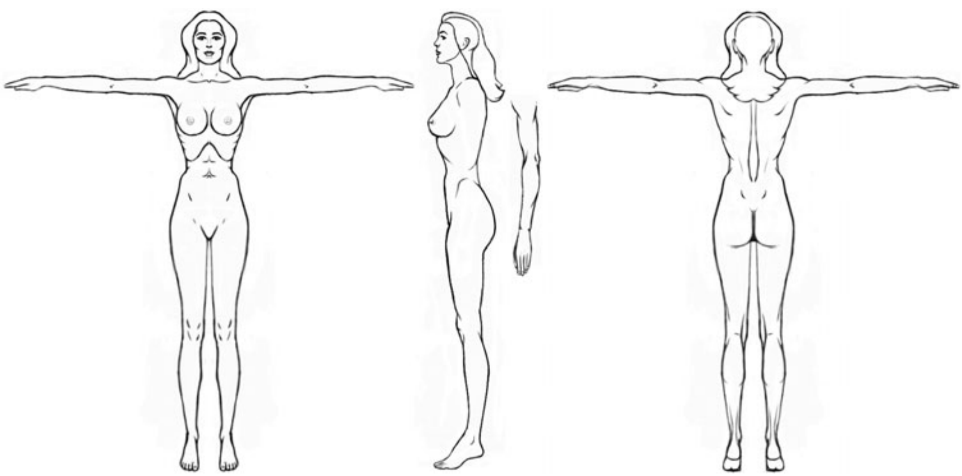


Figure 3.47
The base model sheet for the female character is developed the same way as the male character's base model sheet.

Most female characters have long hair. This creates a problem when you are creating a model because the hair can obscure some areas of the character. To avoid this problem, it is always a good idea to draw the hair as transparent in the model sheet so the modeler can see all areas of the character.

Note

Not all characters are human characters. Often, a character is so unusual that it doesn't make sense to create a base drawing for it. Instead, the artist simply creates the character as a model sheet from the beginning. That said, having a variety of base model sheets is a great timesaver when you are working on new characters. It is a good idea to create base characters for several different body types. Even if a character isn't exactly the same as one of the body types, it is helpful to have something to start with and adapt it to the needs of the new character.

Creating the Template

To make an effective template for building 3D models, each view of the character in the model sheets must be proportional in the 3D program. Each view will be projected onto either a picture plane or a polygon in the 3D program. To make sure the 3D artist can import the model sheets accurately, the concept artist needs to create templates for each view at exactly the same dimensions. Figure 3.48 shows the model sheets converted to modeling templates.

Notice that each drawing is centered in the frame. This is vital so that the modeler is able to line up the drawings correctly. The modeler usually creates templates from the model sheets, but sometimes the concept artist is asked to create them. It is a good idea for the concept artist to understand how to make templates so he or she has a good idea of the importance of making sure every drawing is consistent.

Making Character Model Sheets

Creating character model sheets from a base model is an easy process. By laying a piece of tracing paper over the top of the base model, the artist can create the new character using parts of the base model as a guide. Figure 3.49 shows a warrior character laid over the base model.

The advantage of using the base model sheet is that if characters from that base model sheet have worked well in the past for animation, it is likely that new

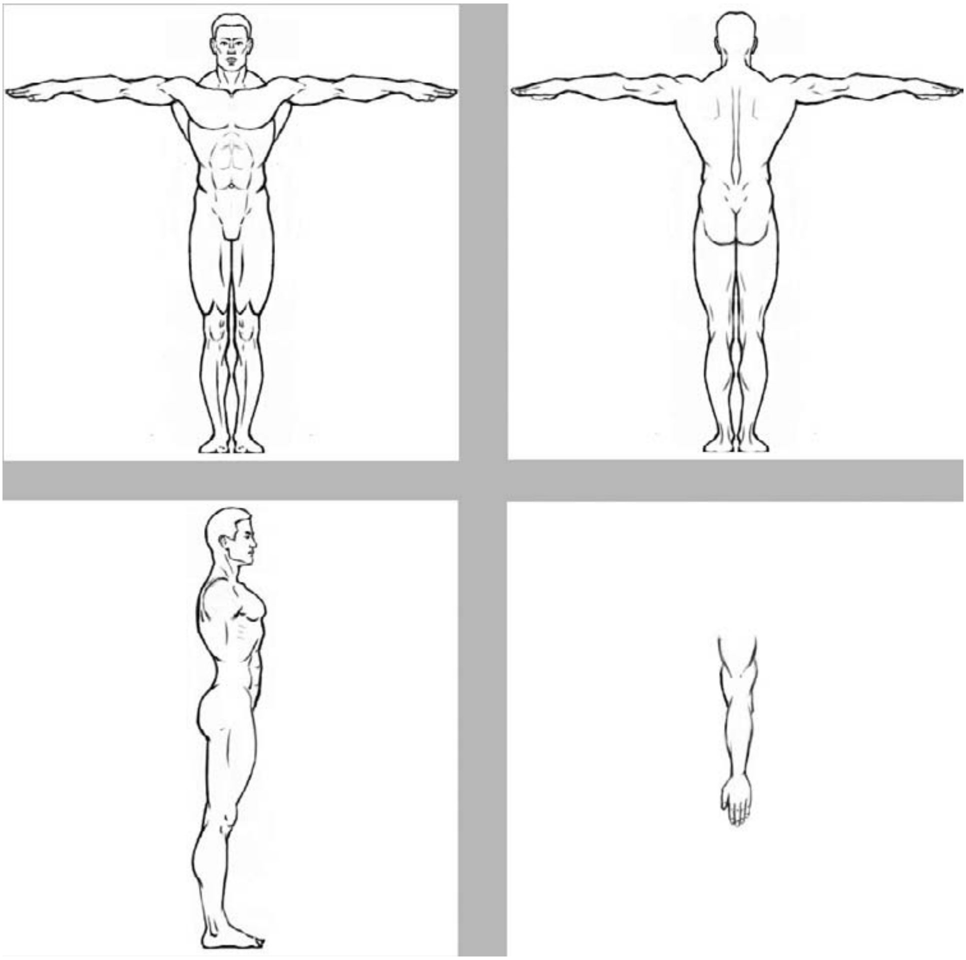


Figure 3.48
Model sheets are converted into templates for 3D modeling.

characters based on that model will work well too. Figure 3.50 shows the finished front drawing of the warrior character.

As shown earlier, you can use the front model sheet to create the rear view of the character. You use a sheet of tracing paper to create the back view of the character from the front drawing. Figure 3.51 shows the back view over the front view.

When you have finished the front and back views, create lines across the paper to line up the back view with the front view and to draw the side view. Figure 3.52 shows the three views together.

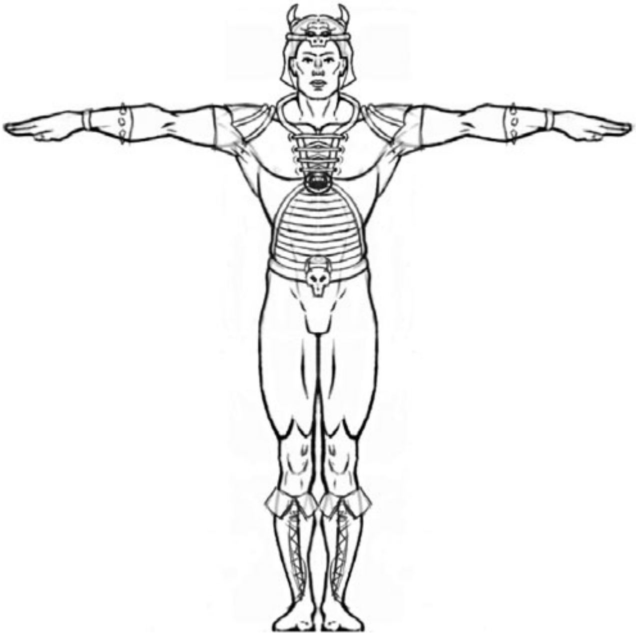


Figure 3.49
The warrior is developed from the base male model sheet.

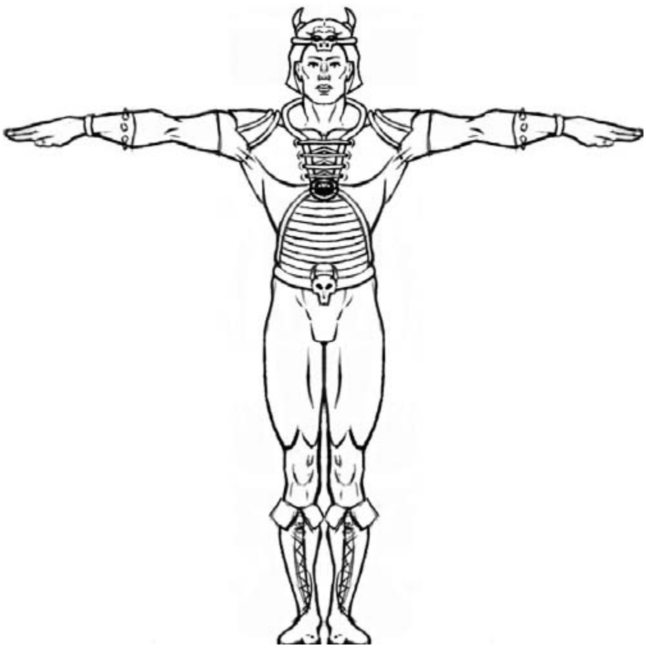


Figure 3.50
The finished warrior character in the front view

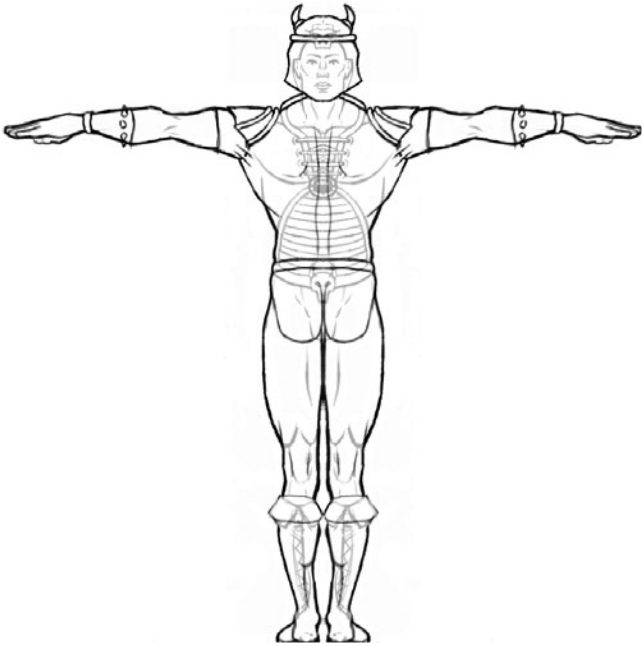


Figure 3.51
Create the back view using the front view as a guide.

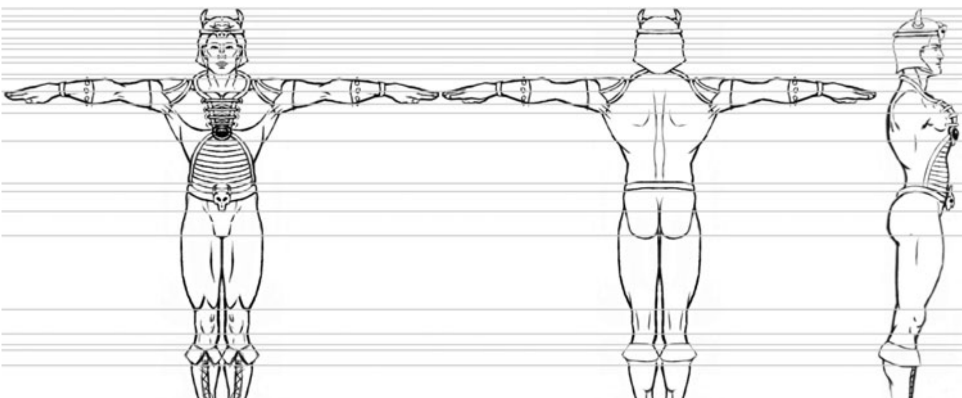


Figure 3.52
Use the guidelines to draw the side view.

To draw the arm correctly, you need to project upward from one of the character's arms, as shown in Figure 3.53.

Now the model sheet of the warrior is complete. Each drawing of the character is accurate. Figure 3.54 shows the finished model sheet without the blue guidelines.

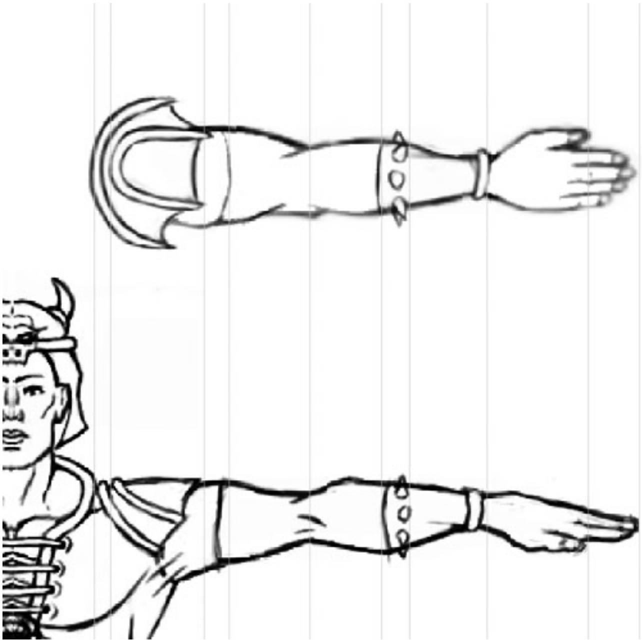


Figure 3.53
Draw the arm by projecting upward.

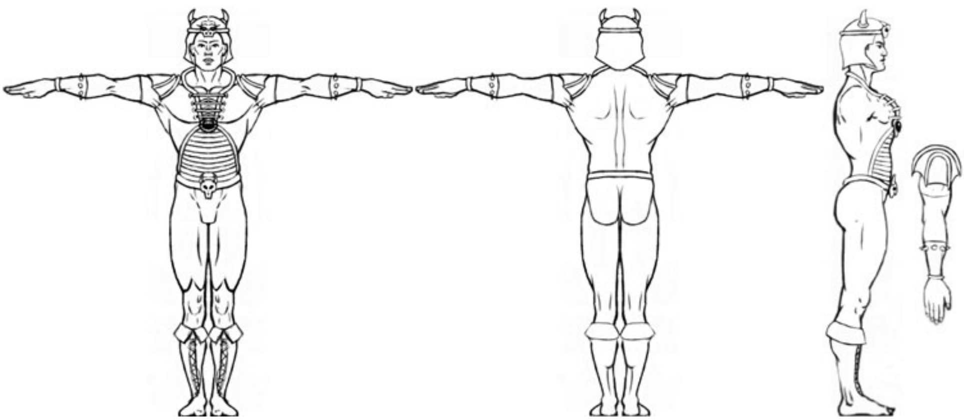


Figure 3.54
Removing the guidelines finishes the model sheet.

This character is symmetrical, but some characters will have differences from one side to the other. If this is the case, you will need to create an extra side view to show the differences from one side to the other. In all circumstances in which a character looks different from one side to another, you should create new views.

Color in Model Sheets

Sometimes a model sheet needs to be in color. This happens when very specific colors are needed for the character. Color in model sheets has the same purpose as any other aspect of the drawings: It gives more direction to the development team. Full-color drawings for model sheets are obviously more expensive and take longer to create than line drawings. They are not created very often because the development team can usually pull the color information from the character illustration. Figure 3.55 shows an example of a full-color model sheet.

Note

The issues regarding accuracy in full-color model sheets are the same as in black-and-white model sheets. The concept artist must be careful not to move details around in the painting process when he or she is developing a full-color model sheet.

Model sheets don't always have to be of characters; sometimes model sheets are created for other game elements. Figure 3.56 shows a model sheet of a house.

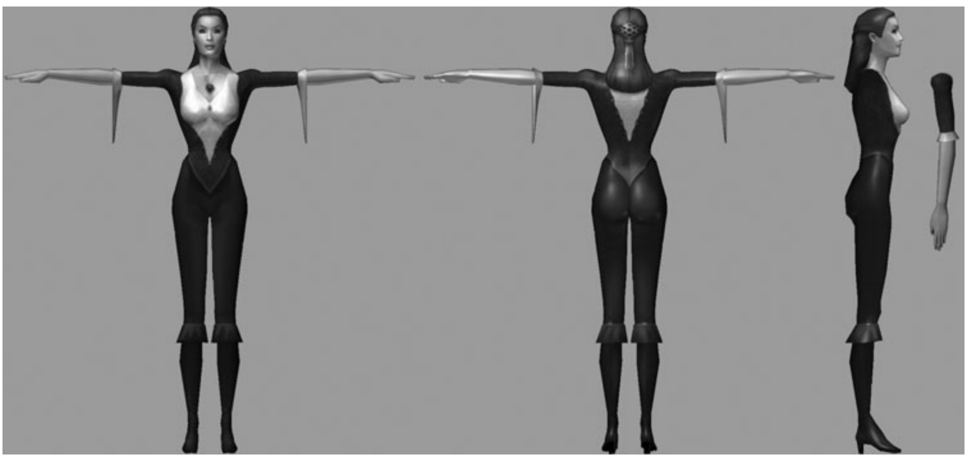


Figure 3.55
Some model sheets need to be in full color.

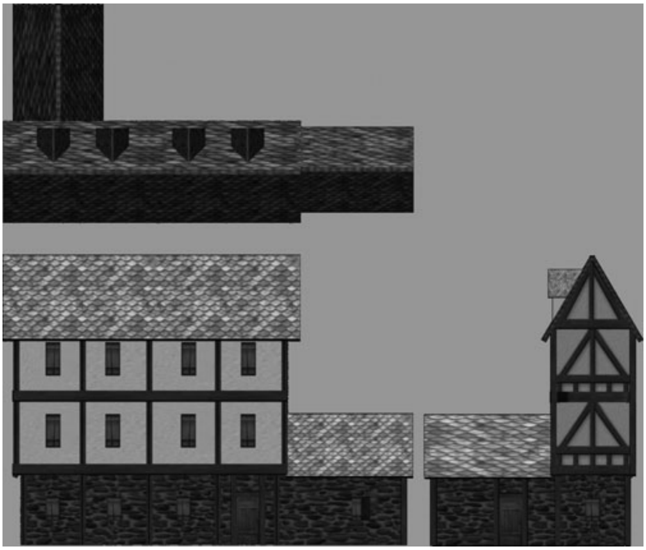


Figure 3.56
Model sheets can be used for other objects in games.

Summary

The last two chapters covered many aspects of character sketches. This chapter dealt with character illustrations and model sheets. A *character illustration* is a detailed, full-color painting of the character. The illustration should be well designed and well rendered because it is used not only for design information, but also for promotion and funding of the game. The character illustration is usually the most refined and finished piece of artwork in the game design.

This chapter also covered several concepts regarding the creation and use of model sheets in game development. The use of model sheets in game development was borrowed from the animated motion picture industry, where the need for consistency in characters is critical. In games, model sheets are used as guides for the development team to create 3D models of characters.

In this chapter, you should have learned the following concepts:

- The difference between a character illustration and sketch
- The definition of a character illustration
- How character illustrations are used in game designs

- What makes a good character illustration
- How to create a simple character illustration
- How to create a detailed, finished character illustration
- What model sheets are
- How model sheets are used in game development
- Why models sheets need to be accurate
- How to create a male base model sheet
- How to create a female base model sheet
- How to create a character model sheet
- How to create a model template

This page intentionally left blank

Chapter 4

Character Modeling

Now that you have laid your foundation, it is time to start with the actual building of a 3D game character. Creating a 3D character for games is done using 3D modeling and animation software. Although there are many great 3D modeling and animation programs, this book focuses on Maya, which is the 3D program used in many of the major studios. It is widely considered one of the best programs for advanced modeling and animation in the industry. In addition to the full version, Maya also has a full-featured learning edition called Maya PLE, which is included on the CD that comes with this book.

Installing Maya

Before you can start using Maya PLE to create the projects in this book, you will need to install it on your computer. Maya is a professional program, and as such, does not work with earlier version of Windows. You will need to have Windows 2000 Professional or either the Home or Professional version of Windows XP.

Note

The software calls for XP Professional, but it will run adequately on the Home version for the projects in this book.

1. Put the CD that accompanies this book in your CD-ROM drive and wait for the menu to appear.
2. On the menu, select Install Maya PLE and follow the onscreen directions.

3. The program will ask for a key to finish installation; you must obtain this key from AutoDesk. The installation program will instruct you on how to receive this key in an e-mail from AutoDesk.
4. When you have finished installing Maya PLE on your computer, proceed to the next section of this chapter to familiarize yourself with the software.

Note

The next section of this chapter deals with the basic layout and functions of Maya. If you are already familiar with the software, you can skip ahead to the latter part of the chapter.

The Maya Screen

At first look, the Maya interface can be a little overwhelming. The sheer number of menus and buttons seems endless, and it would take volumes of books to explain every feature in Maya. Don't worry right now about learning every feature. The step-by-step instructions in this book are designed to help you understand the features of the program that are needed to create each project. As you work with Maya, you will begin to become accustomed to the features.

The first thing that you need to know about Maya is how the program is set up and where the essential tools are located. From there, you can move on to creating 3D digital art. The more you use Maya, the more accustomed you will become to the tools.

Figure 4.1 shows the Maya interface as it appears the first time it is displayed.

The interface can be broken down into several components:

- The workspace
- The toolbox and Quick Layout buttons
- The main menu
- The Status line
- Shelves
- The sidebar
- The animation tools
- The Command and Message lines

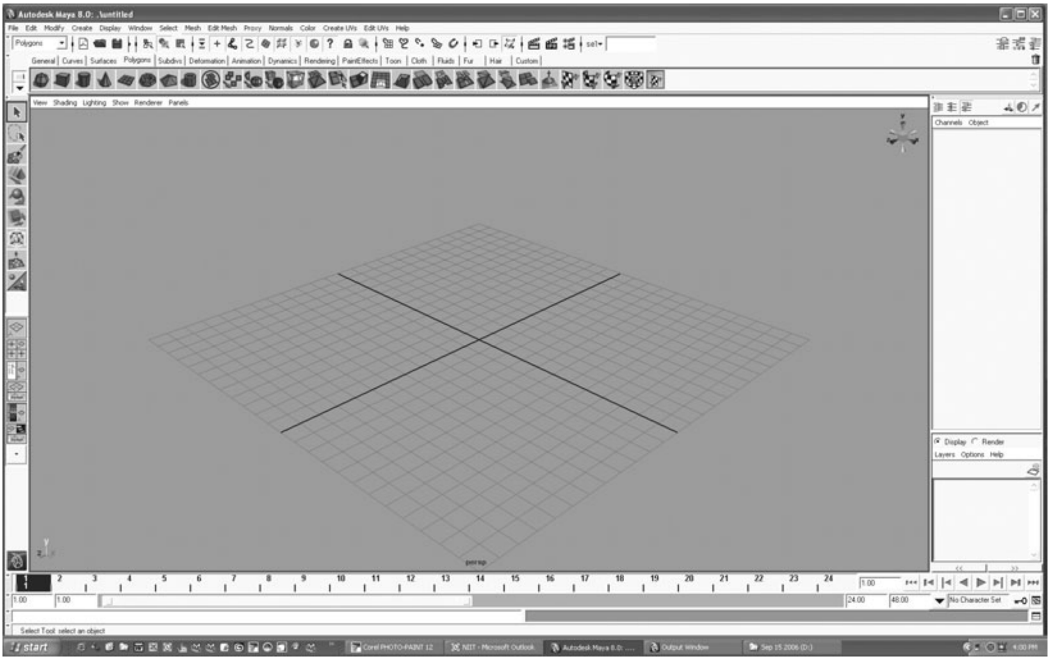


Figure 4.1
The Maya interface screen has many features and options.

The Workspace

The workspace, shown in Figure 4.2, is where most of your work will take place. It is where you build the model, animate your characters, apply textures, and just about everything else.

There are a few important elements of the workspace that you should know about:

- The Panels menu is used to control functions and features specific to the workspace.
- Within the workspace is a grid with two intersecting dark lines. The point at which these two lines intersect is called the *origin*. The origin is the zero point of the X, Y, and Z axis.
- In the upper-right and lower-left areas of the screen are two helpful little indicators. These are the Axis Direction indicators.

In addition, you'll want to familiarize yourself with Maya's workspace views, as well as how to navigate the workspace.

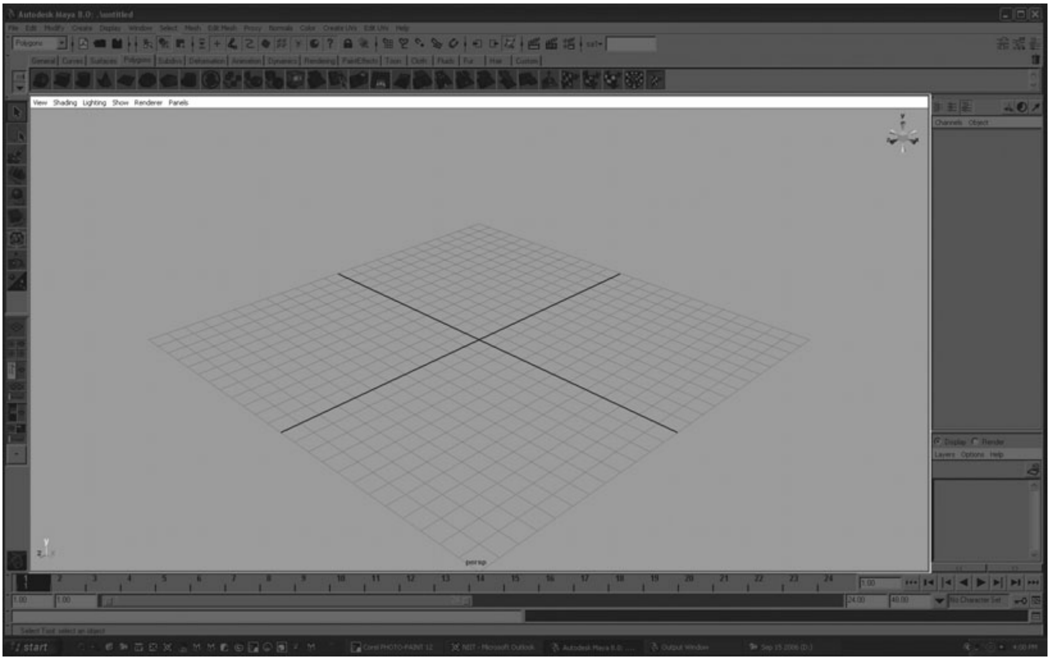


Figure 4.2
The workspace is the work area of the interface.

Changing the View

Imagine a director shooting a movie with multiple cameras, looking at the scene from several different angles. Depending on which camera is used, the scene will look different. Likewise, Maya has several *virtual* cameras. Each camera displays a different camera view, or just view for short. The name of the camera whose view is currently displayed in the workspace is located along the bottom of the workspace, in the center.

You can configure your workspace to have a single view, as is the case the first time you launch Maya, or many views. To set the number of views, do the following:

1. Open the Panels menu and choose Layouts, as shown in Figure 4.3
2. Choose Two Panes Side By Side. The workspace changes to contain two views, each view displayed in a *pane*.
3. Experiment with some of the different layouts to see what they look like. When you are finished, return to the Single Pane view. A two pane split of the workspace is shown in Figure 4.4

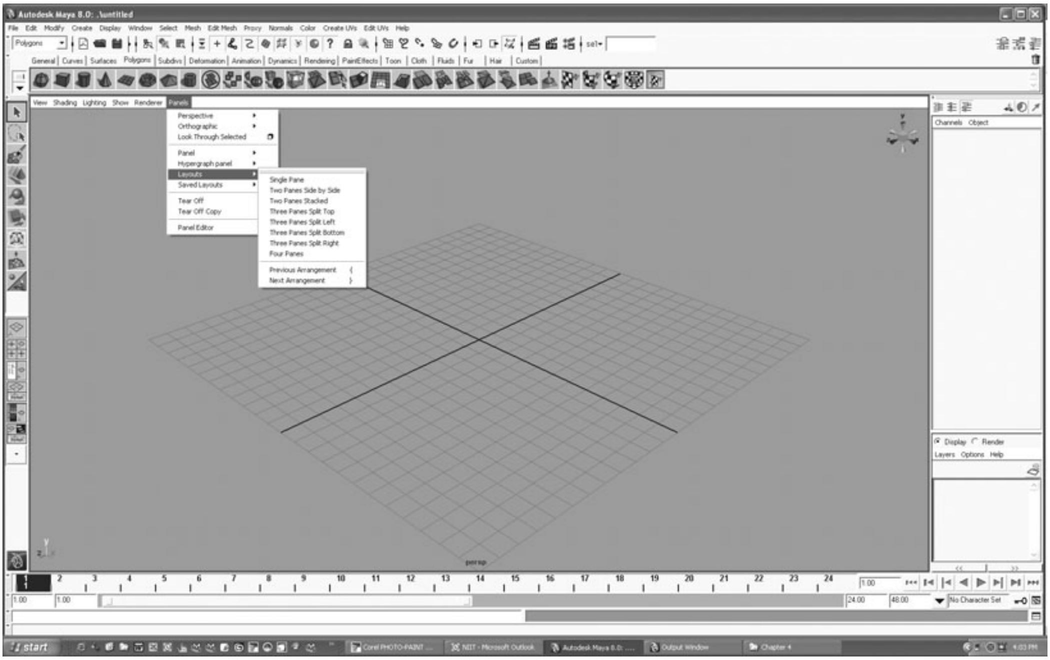


Figure 4.3
There are several preset panel layouts in Maya.

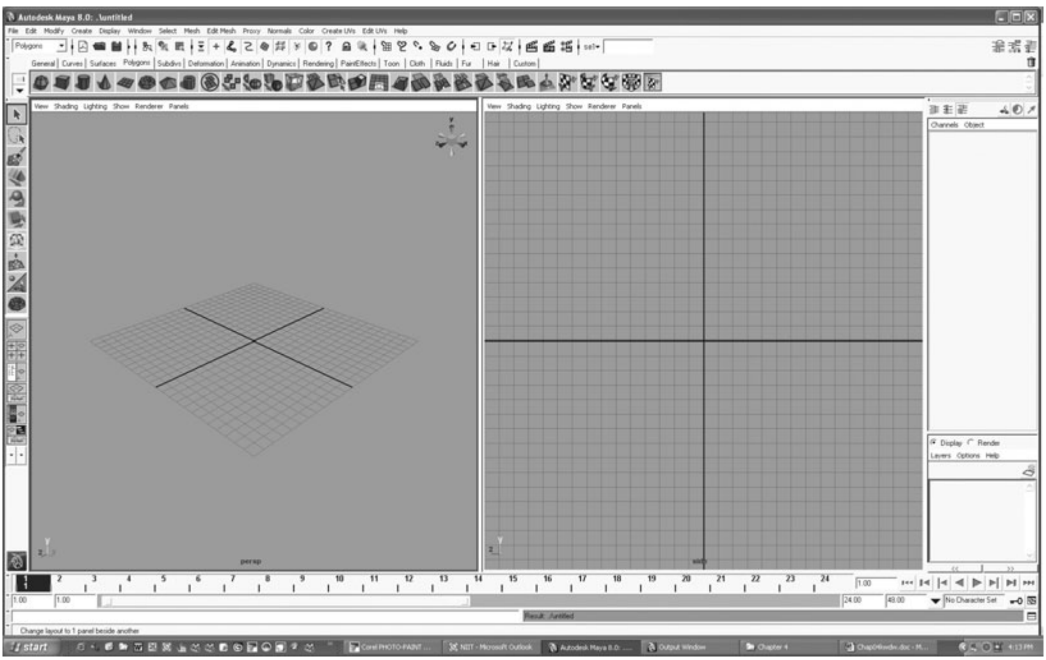


Figure 4.4
Two view panes in the workspace.

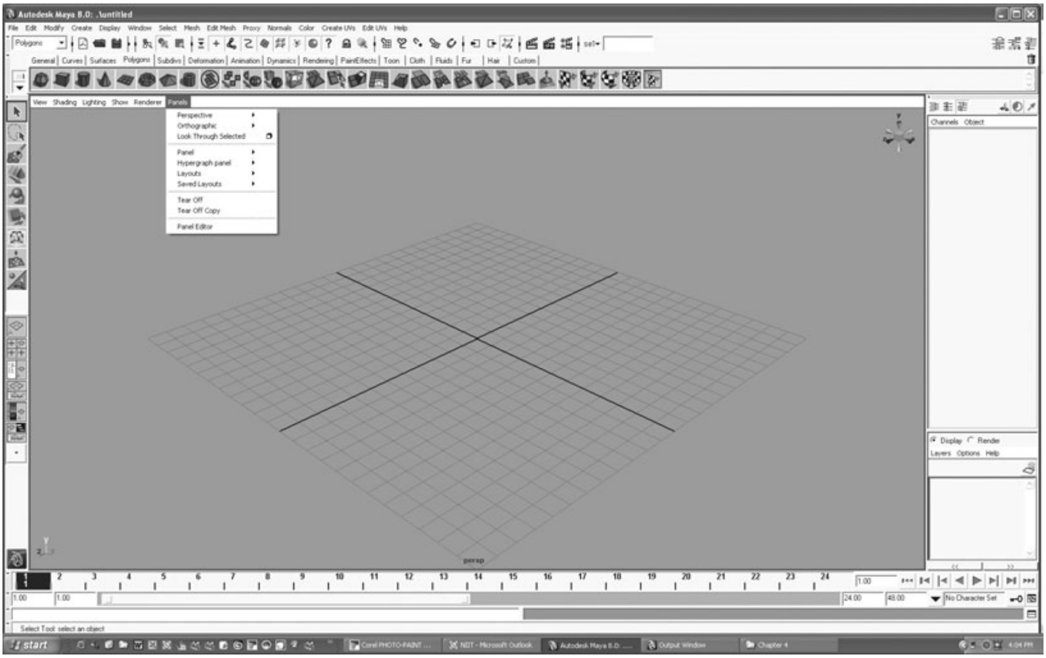


Figure 4.5
Perspective and Orthographic are the two types of views, found at the top of the Panels menu.

In addition to configuring how many views are displayed at once, you can also specify what *type* of view is displayed. As shown in Figure 4.5, there are two options:

- **Perspective.** Perspective views show objects as if you were looking at them in real life, with perspective.
- **Orthographic.** Orthographic views are like drafting views. They show the model straight on with no perspective.

Try creating an object so you can better see how these views work. Do the following:

1. Open the Panels menu, choose Orthographic, and select Front.
2. Click Create in the main menu to open the Create menu.
3. Choose Polygon Primitive and select Torus and click in the workspace to place the torus.
4. Open the Panels menu, choose Layout, and select Four Panes to change the layout to four panes, each pane with one camera view, as shown in

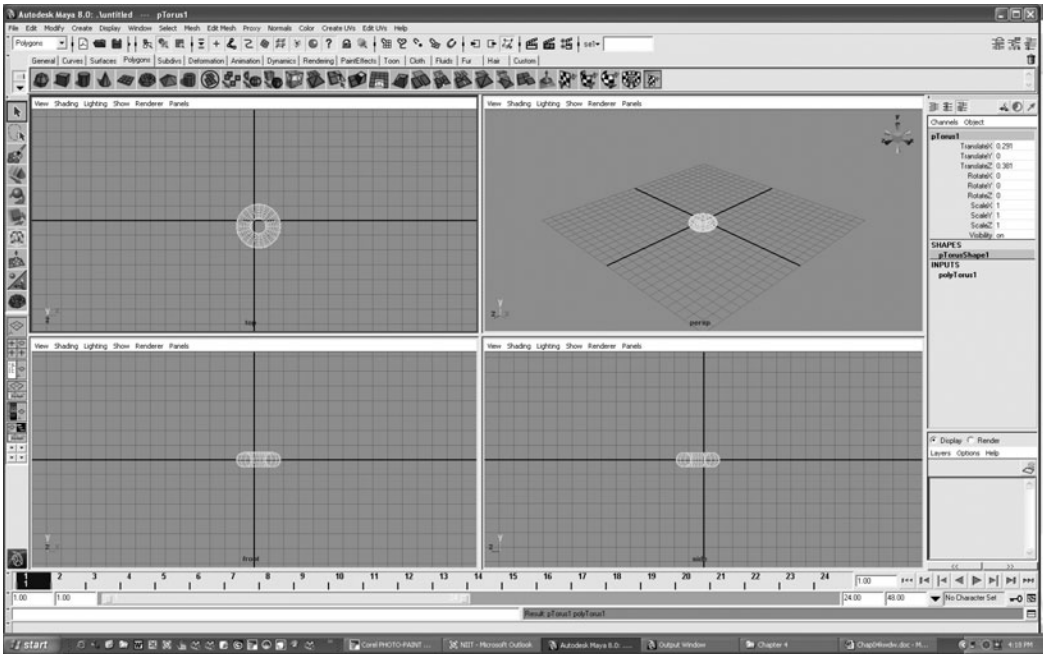


Figure 4.6
Change the workspace layout to four panes.

Figure 4.6. You can see from the various views that the object you created looks very much like a doughnut.

Note

When you change the workspace to four panes, whatever view you started in will be in the upper-left pane. Because you started in the front view here, you currently have two front views. You can change any pane to a new view from the Panels menu; there is a Panels menu for each pane.

Navigating the Workspace

Now that you know how to change views and layouts, let's take a look at navigating within the workspace. Do the following:

1. Open one of the Panels menus, choose Layout, and select Single Pane.

2. Open the Panels menu and choose Perspective.

3. In Maya, you use the Alt key for navigation. While pressing the Alt key and then hold down the left mouse button. You will see the cursor change to two arrows making a circle.

4. Move the mouse around a little. Notice how easy it is to look at the torus from almost any angle. In Maya, this action is called *tumbling*.
5. While pressing the Alt key, hold down the center mouse button or mouse wheel (if your mouse has one). The cursor will change to a circle with for arrows extending from it.
6. Move the mouse around. Notice that the view is panning. In Maya, this is called *tracking*.
7. In addition to tumbling and tracking in the workspace, you can also dolly. *Dolly* is an old cinematic term from when cameras were mounted on tripods with wheels so that the camera could be moved in and out of a scene. To dolly in Maya, while pressing the Alt key, hold down the right mouse button.
8. Move the mouse down or to the right to zoom into the scene; move the mouse up or to the left to pull back from the scene.

The Toolbox and Quick Layout Buttons

Figure 4.7 highlights the area in the far left of the screen. This area houses the toolbox and the Quick Layout buttons. The toolbox, located in the upper portion of the highlighted area, it contains model-manipulation tools, with each tool represented by a button. The lower part of the area features buttons that are used to change the view layout of the workspace.

There are ten tools in the toolbox; each is used to either select or modify an object or part of an object. The tools going from top to bottom are as follows:

- **Selection tool.** Use this tool to create a selection by clicking or by drawing a bounding box.
- **Lasso tool.** Use this tool to create a selection by drawing a defined area.
- **Paint selection tool.** Use this tool to select parts of an object by painting on the object.
- **Move tool.** Use this tool to move, or *translate* (that is, move an object from one location to another), a selected object or component in the scene.
- **Rotate tool.** Use this tool to rotate a selected object or component in a scene

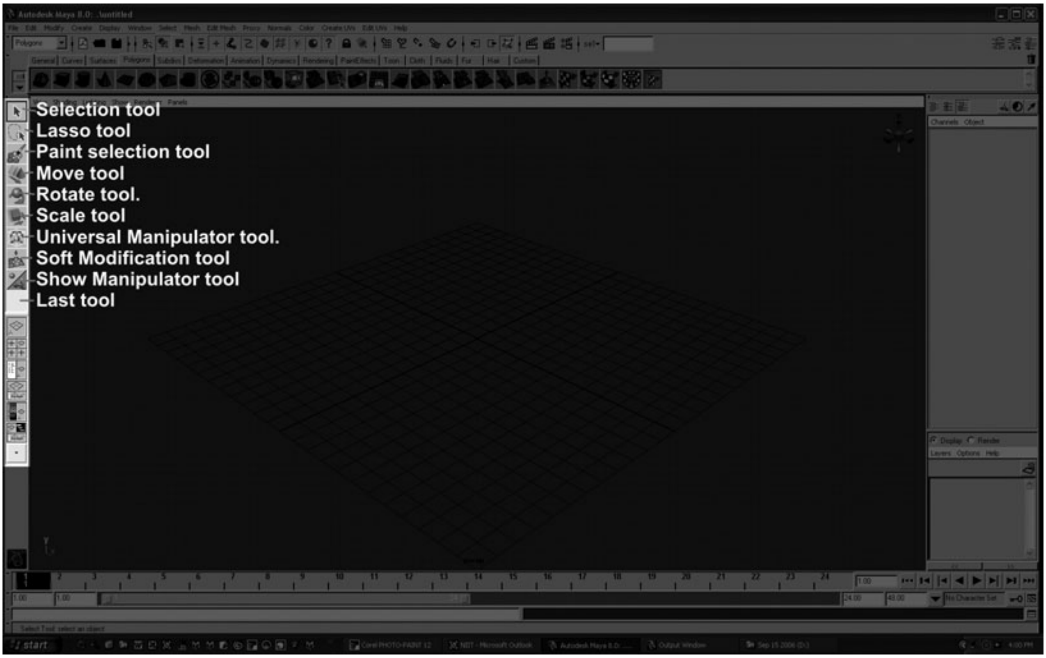


Figure 4.7
The highlighted area contains the toolbox and the Quick Layout buttons.

- **Scale tool.** Use this tool to scale up or down in size a selected object or component.
- **Universal Manipulator tool.** This tool combines the Move, Rotate, and Scale tools into a single tool.
- **Soft Modification tool.** This tool is used to soft-modify an object. (To soft-modify an object is to adjust its vertices, similar to how a sculptor modifies a lump of clay.
- **Show Manipulator tool.** Some elements, like lights, have their own manipulators. Clicking this tool activates the manipulator for an element.
- **Last tool.** The Last tool is not really a tool, but rather holds the last tool or function used. This becomes very handy when you have a repeating function.

Below the toolbox are the Quick Layout buttons. You can change the workspace layout to one of a few widely used configurations by simply clicking one of these buttons.

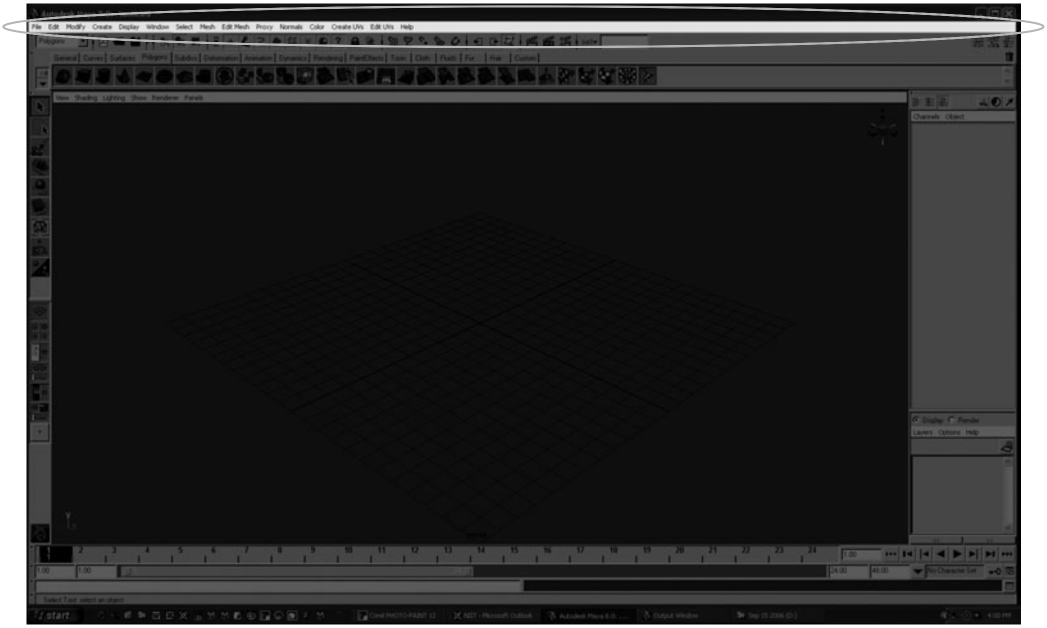


Figure 4.8
Many of Maya's features are located in the main menu.

The Main Menu

Figure 4.8 shows the main menu, where you access many of Maya's features. Almost every function or tool can be found in the main menu or one of its submenus. There are several configurations of the main menu depending on the type of work being performed; the next section covers changing the menu configuration.

The Status Line

Figure 4.9 shows the Status line, which contains many buttons. On the far left side of the Status line is a drop-down list that you can use to configure the main menu. The available configurations are as follows:

- Animation
- Polygons
- Surfaces

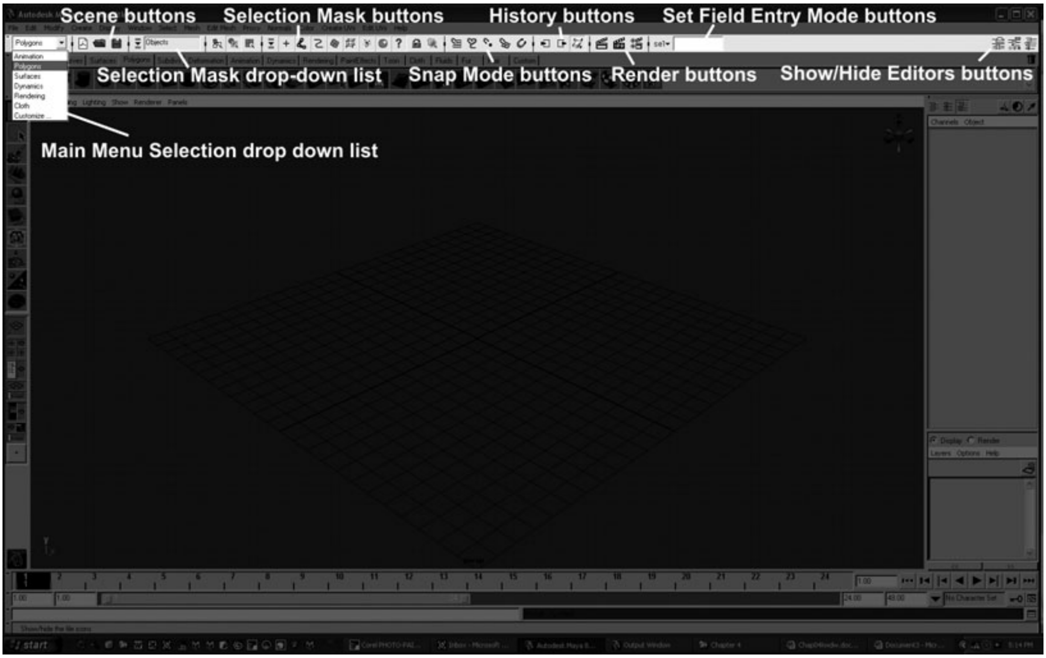


Figure 4.9
The Status line is directly below the main menu.

- Dynamics
- Rendering
- Cloth
- Customize

The Status line also contains many common functions. If you hover the cursor over any button or icon on the status line, a window will appear identifying the button or icon. These functions are split into groups as follows from left to right:

- **Scene buttons.** These are used to bring up new scenes or to save and load scenes.
- **Selection Mask drop-down list.** This is used to change the selection mask so that it is easier to select a particular element in the scene.
- **Selection Mask buttons.** These are used to help with selections by masking objects of components.

- **Snap Mode buttons.** These used for selecting different modes of snapping.
- **History buttons.** These are used to enable, disable, and access the construction history.
- **Render buttons.** These are used to render the scene and set rendering parameters.
- **Set Field Entry Mode buttons.** These are used to change the field entry.
- **Show/Hide Editors buttons.** These are used to show or hide editors and tools used to modify or manipulate objects or parts of objects. These editors appear in what is called the *sidebar,* which is the area directly to the right of the workspace.

The Shelves

In the area shown in Figure 4.10, directly below the Status line, are the shelves, which contain buttons and tabs. Each tab has different buttons, and each button has a different function in Maya. The tabs are used to group functions for specific

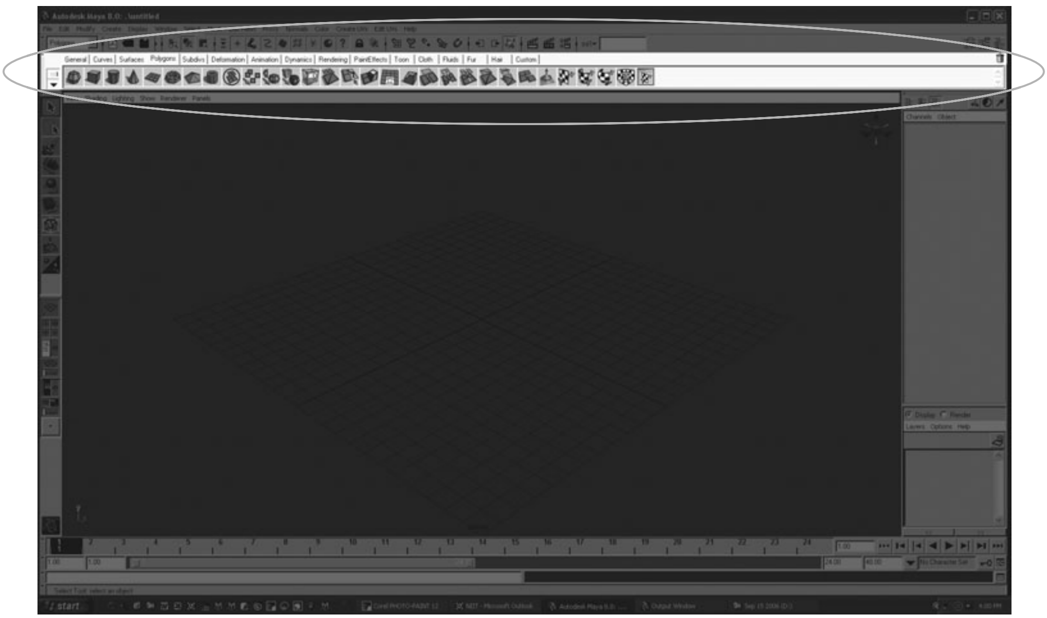


Figure 4.10
The shelves contain buttons that call up specific functions.

tasks. The tab shown in Figure 4.10 is the Polygon tab. Click a few other tabs to see how they look.

Note

You can build your own shelf if you want to; click the black arrow on the far left of the shelves to display several editing functions for the shelves, including the Shelf editor.

The Sidebar

Figure 4.11 shows the sidebar. This area contains several editor elements and consists of two parts:

- **The Channel box.** The Channel box, located in the top portion of the sidebar and often used in modeling and animating in 3D, is used to edit object attributes such as the object's position, rotation, scale, and so on. Using the Channel box is the way to go when you need an attribute to be assigned a specific numeric value. Below the attribute list in the

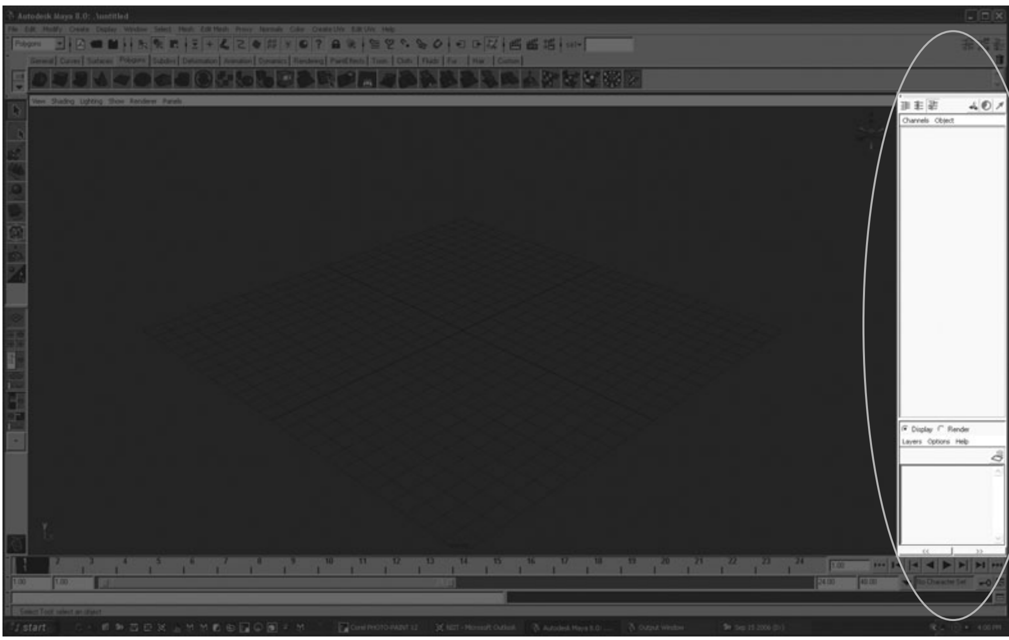


Figure 4.11
The Channel box and Layer editor are part of the sidebar.

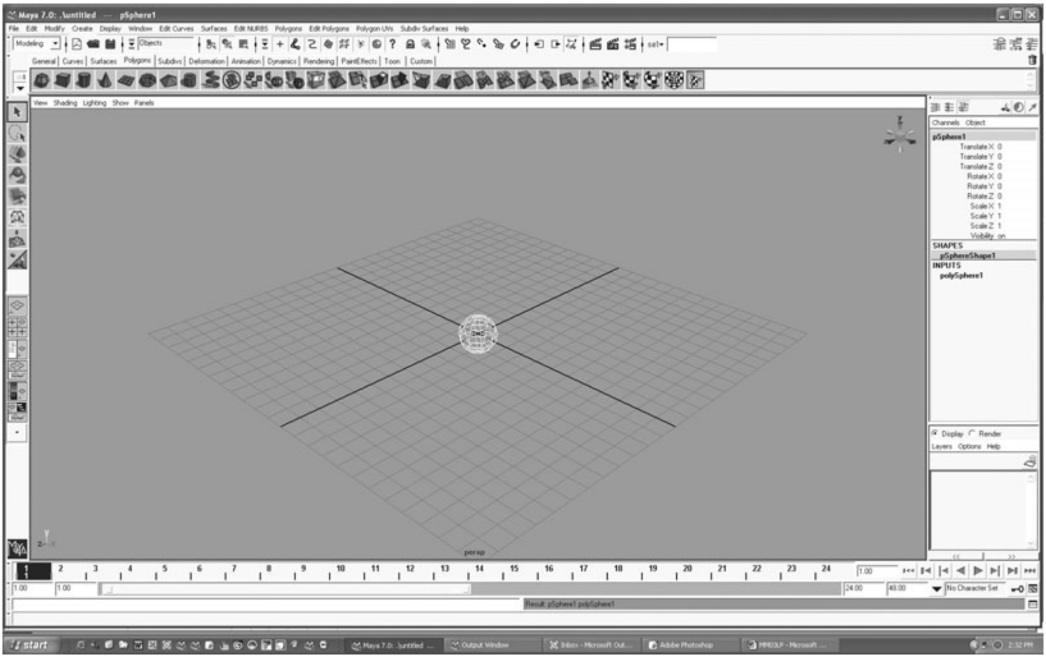


Figure 4.12
The Channel Box is used to change object attributes.

Channel box is a history, which contains a list of all modifications made to the object. Figure 4.12 shows the Channel box for a polygon sphere object.

- **The Layer editor.** Found in the bottom portion of the sidebar, the Layer editor is used to define objects in the workspace as layers so that those objects can be isolated in groups from the rest of the scene. The Layer editor is used to organize scene elements, and is particularly useful in complex scenes. For example, a city scene might be organized into roads, buildings, foliage, and vehicles. Each type of scene element is put on a different layer, enabling the artist to hide or reveal each type of element simply by turning its layer on or off.

The Channel box and Layer editor are displayed in the sidebar by default, you can display other tools in this area if you prefer. To do so, use the three buttons on the far right side of the Status line. Clicking the button on the right shows or hides

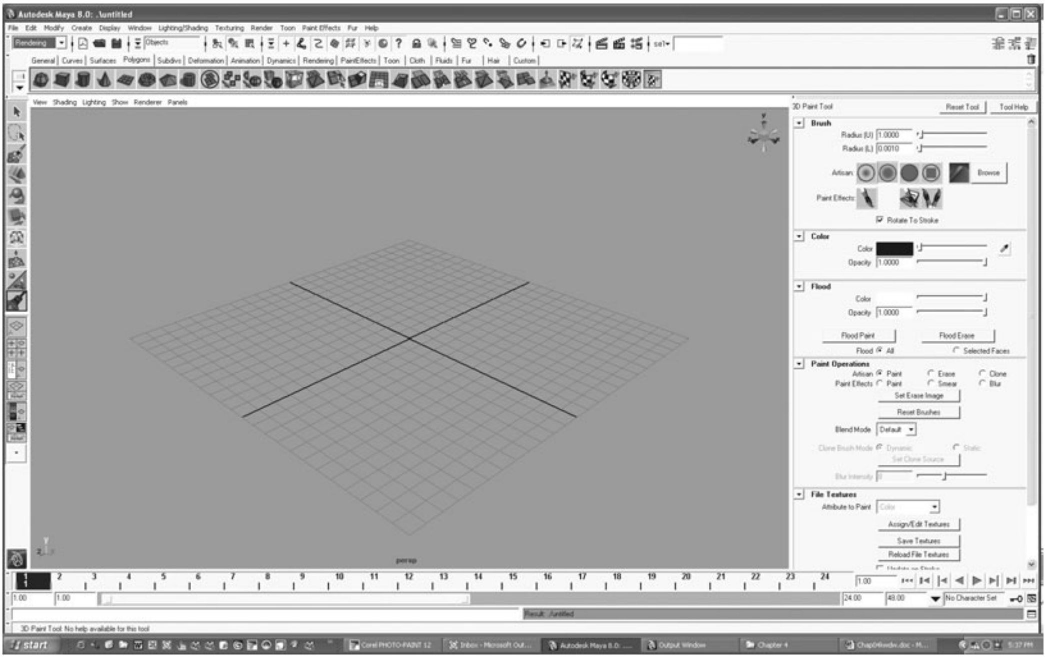


Figure 4.13
The painting tools appear in the sidebar.

the Channel box. Clicking the middle button shows or hides the Tool Options pane. (The Tool Options pane contains specific options for many of Maya's tools. Figure 4.13 shows the Sculpt Geometry tool, one of a number of tools that contain multiple options. Using the Tool Options pane, the artist can set and change options for the tool.) Clicking the button on the left shows or hides the Attribute editor. (The Attribute editor is used to edit attributes of a single node. A *node* holds specific information about a construct in Maya. Figure 4.14 shows a Lambert material; all editable aspects of the Lambert material are contained in the Attribute editor.)

Animation Tools

Figure 4.15 highlights the part of the screen where Maya's animation tools are located. These include the Time slider on the upper portion of the highlighted area and the Range slider on the lower portion of the highlighted area. The right side of the highlighted area contains several tools for playing and

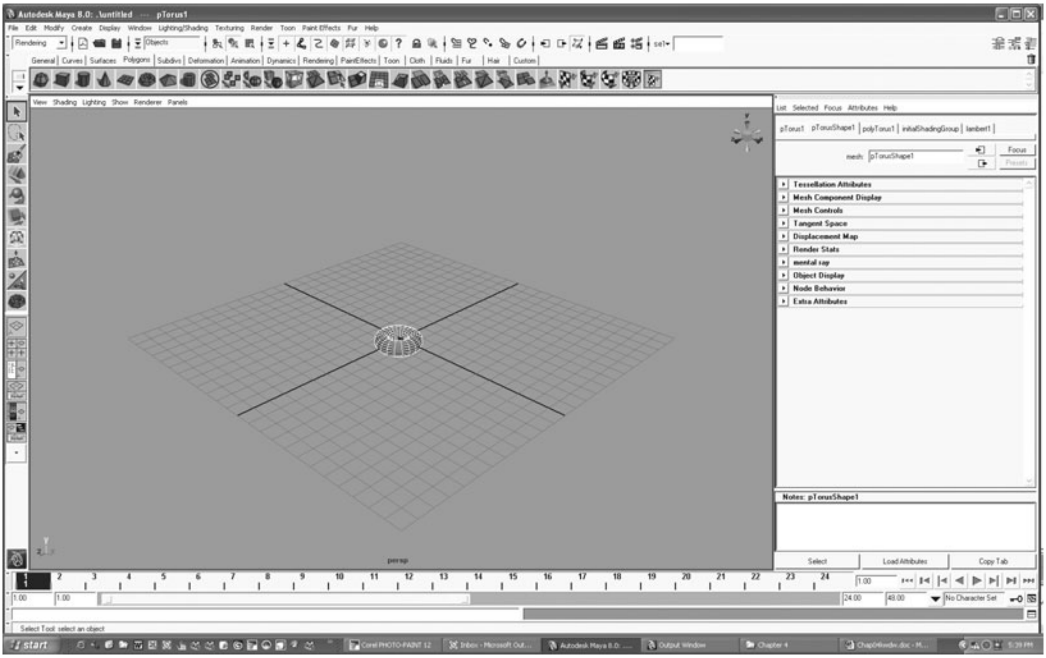


Figure 4.14
The Attribute editor is a tool for editing attributes of a single node.

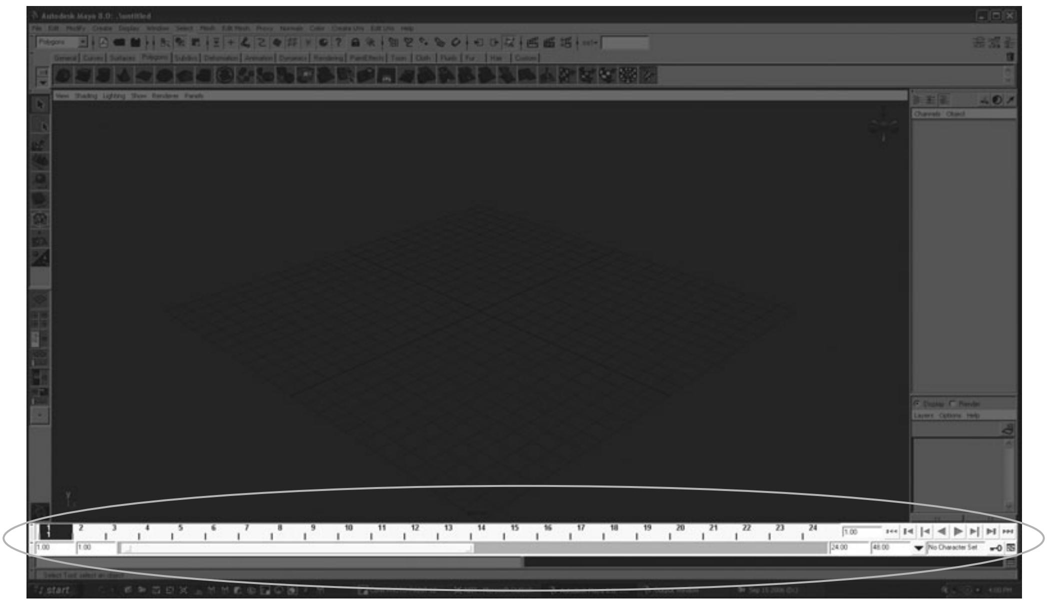


Figure 4.15
Maya's animation tools are located in the lower part of the screen.

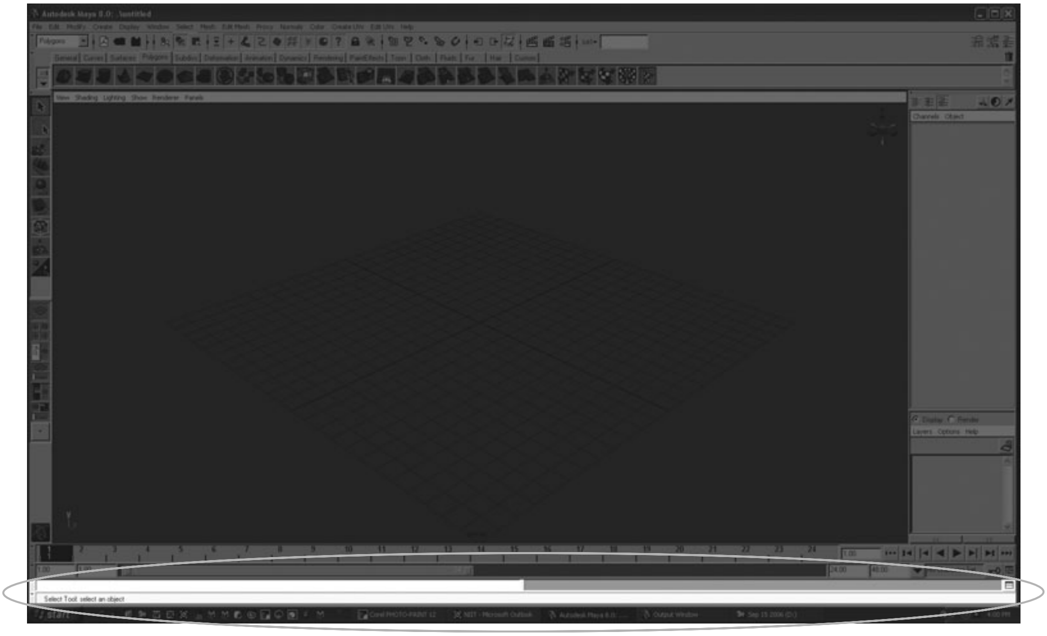


Figure 4.16
The Command and Help lines are located at the bottom of the screen.

viewing animations like the Auto Key, Animation Preferences, and Playback controls.

Command and Message Lines

At the bottom of the Maya screen, as shown in Figure 4.16, are three text windows. The one on the upper-left side of the highlighted area is the Command line, which is where special scripts can be entered. The text window to the left of the Command line shows the results of each command. Below the Command line is the Help line, which is used to display help messages. On the far right is a button that displays the Script editor.

See, that wasn't so bad. Now that you know the basic parts of the Maya screen, it's time to put some of that knowledge to work. Click the New Scene button located just right of the Main Menu Selection drop-down list on the Status line to reset the scene.

Having a Ball

This section is very basic, so if you have experience with 3D programs you may want to skip it. If, however, you are new to Maya, this section will help you get started. In this section you learn to create a simple 3D ball.

1. Create a polygonal sphere, as shown in Figure 4.17. To do so, open the Create menu, choose Polygon Primitive, and click the box next to the Sphere command.

Note

The box next to the Sphere menu entry indicates that there is a dialog box associated with the function. Dialog boxes are used to change attributes or settings for the function.

2. The Polygon Sphere Options dialog panel opens. Change the Radius setting to 10.

3. Change the Subdivisions Around Axis and Subdivisions Along Height settings to 20.

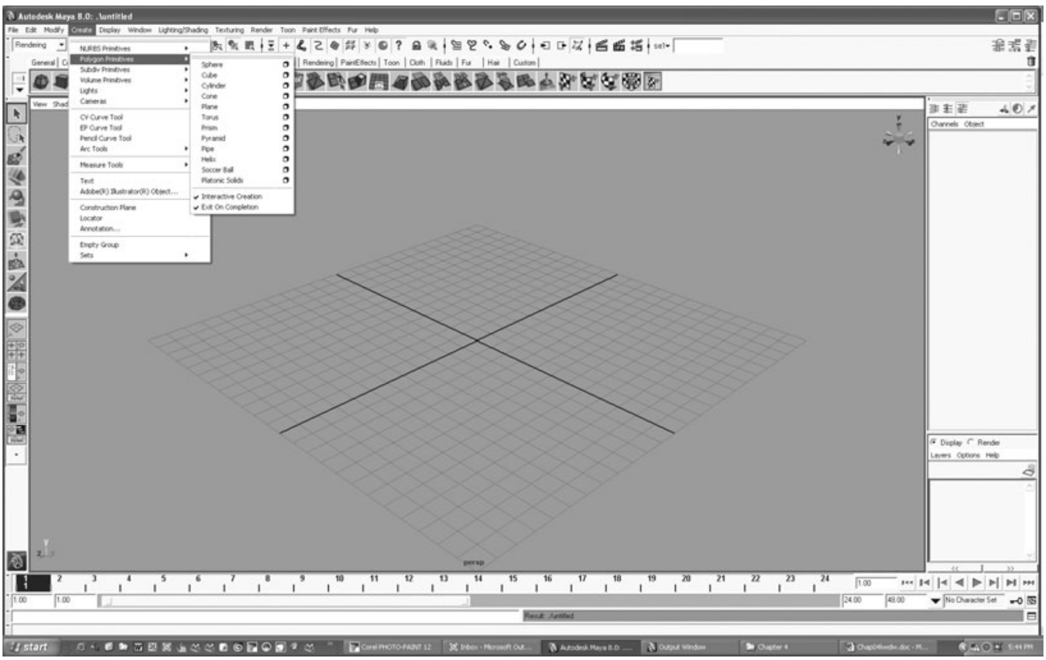


Figure 4.17
Select the box next to the Sphere option.

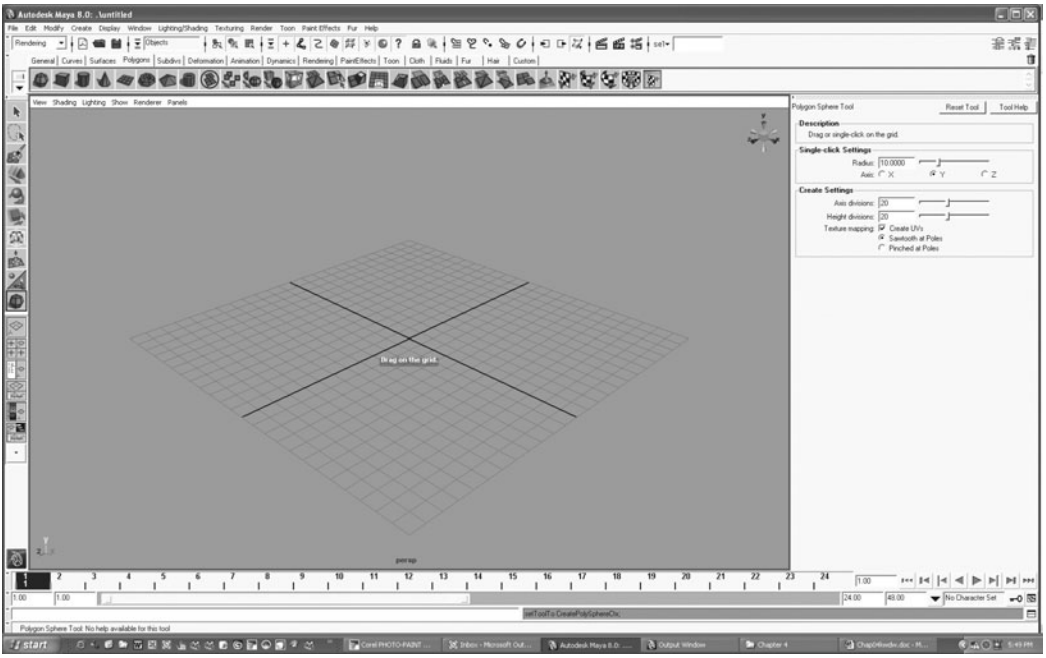


Figure 4.18
Use this dialog box to set the options for creating a polygonal sphere.

4. Change the Axis setting to Y.

5. Change the Texture setting to Saw Tooth at Poles. The dialog box should match the one in Figure 4.18.

6. Click the Create button to create the sphere; the results are shown in Figure 4.19.

7. The 4, 5, and 6 keys on the keyboard control the view modes in Maya. The 4 key changes the view mode to Wire Frame; the 5 key changes the view mode to Flat Shaded; and the 6 key changes the view mode to Smooth Shaded. (The view modes can also be controlled using the View option in the Panels menu.) Press the 6 key on the keyboard to change to Smooth Shaded view; Figure 4.20 shows the result.

8. The next step is to change the surface of the sphere. Maya has a tool called Hypershade that is used to change the surfaces of objects. To access this tool, open the Window menu, choose Rendering Editors, and select Hypershade, as shown in Figure 4.21.

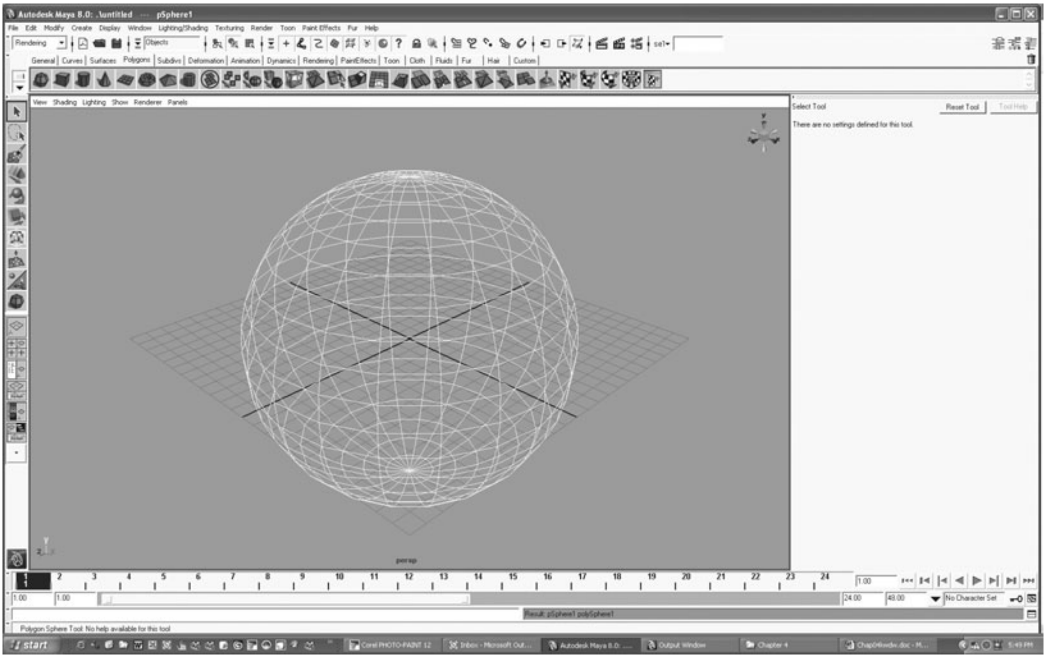


Figure 4.19
The new polygonal object is now in the work area.

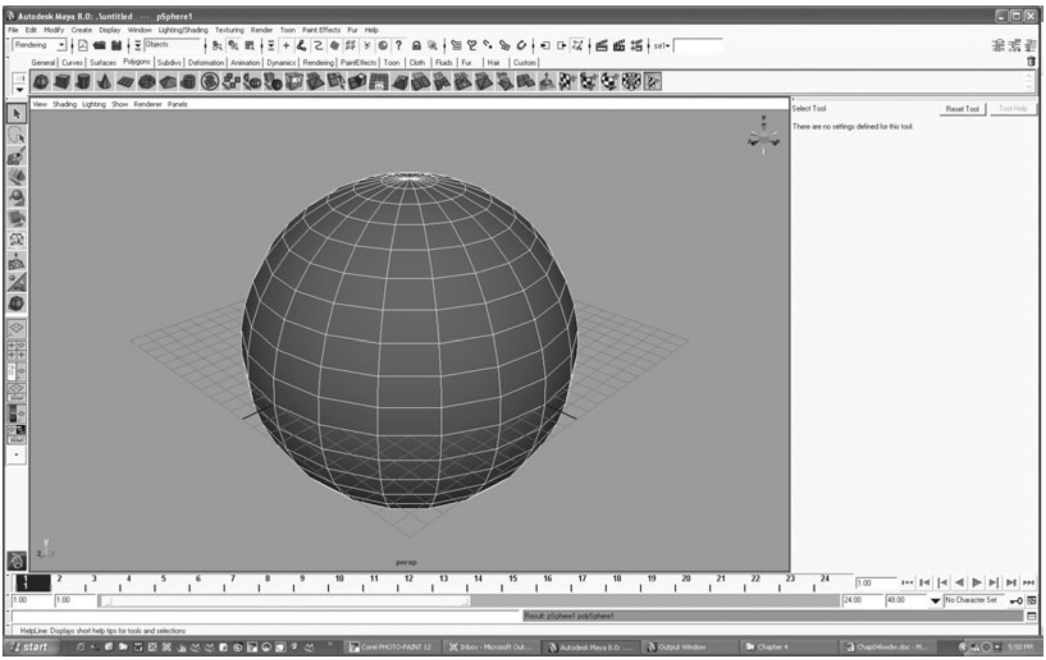


Figure 4.20
Change the view mode to Smooth Shaded.

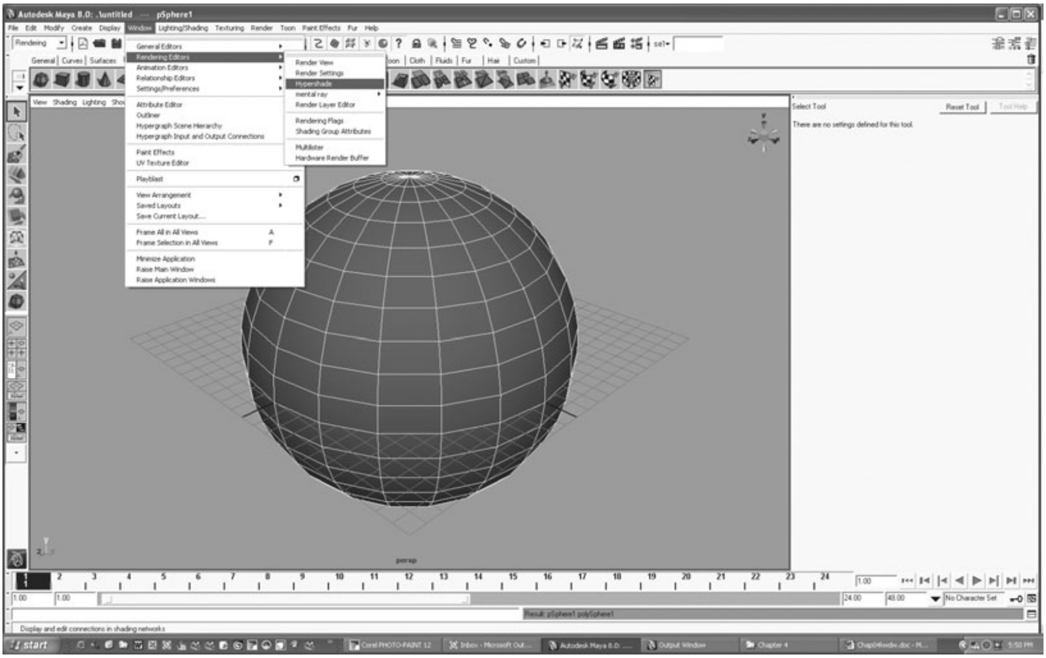


Figure 4.21
Hypershade is one of the options in the Rendering Editors submenu.

9. In the Hypershade window, open the Create menu, choose Materials, and select Blinn, as shown in Figure 4.22.

10. Blinn materials have a number of attributes that can be adjusted to create sophisticated surfaces on 3D objects. To do so, you use the Attribute editor. To launch the Attribute editor, click the Attribute Editor button in the Status line (it's the button that's third from the end). The Attribute editor opens in the sidebar, as shown in Figure 4.23.

11. Click the checkered button to the right of the Color setting in the Attribute editor to open the Create Render Node window, shown in Figure 4.24.

12. The Create Render Node window features many choices for changing the color and texture of a material; click the Checker button. The material in the Hypershade window assumes a checkered pattern.

13. Apply the material to the sphere. To do so, select the sphere in the workspace, right-click the new material in Hypershade to open the marking menu, and choose Apply Material to Object. The sphere now has a checkered pattern, as shown in Figure 4.25.

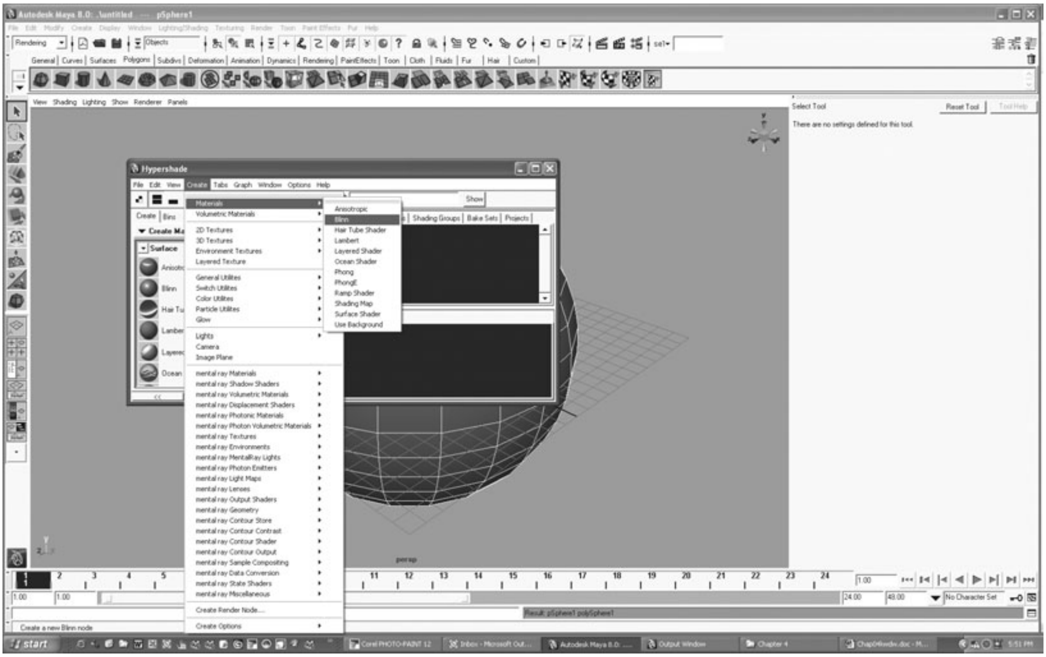


Figure 4.22
Create a Blinn material.

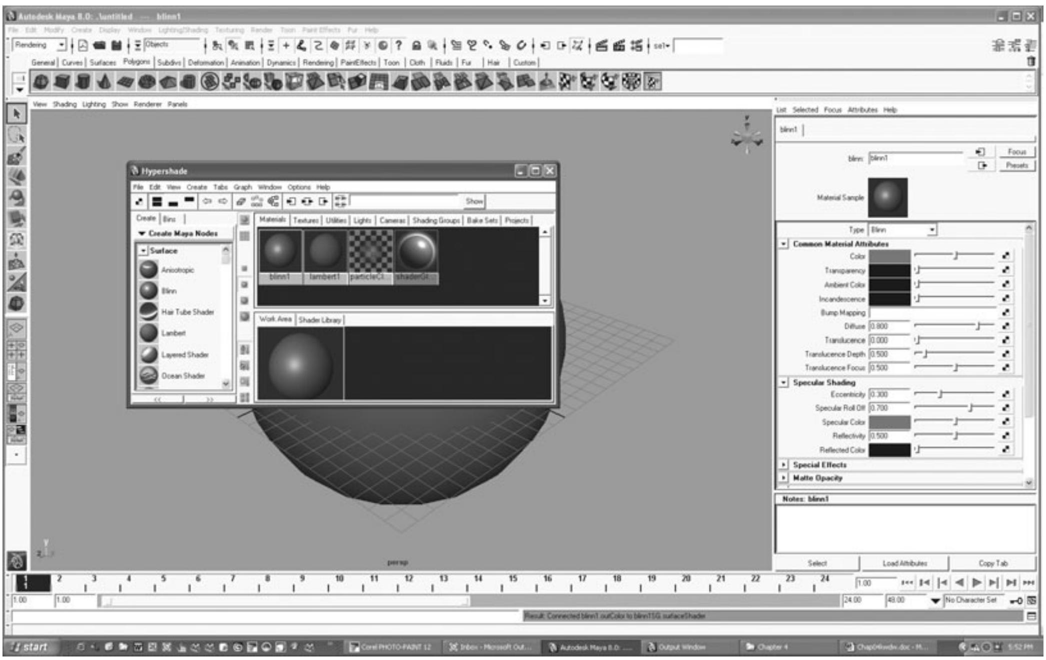


Figure 4.23
Open the material's Attribute editor.

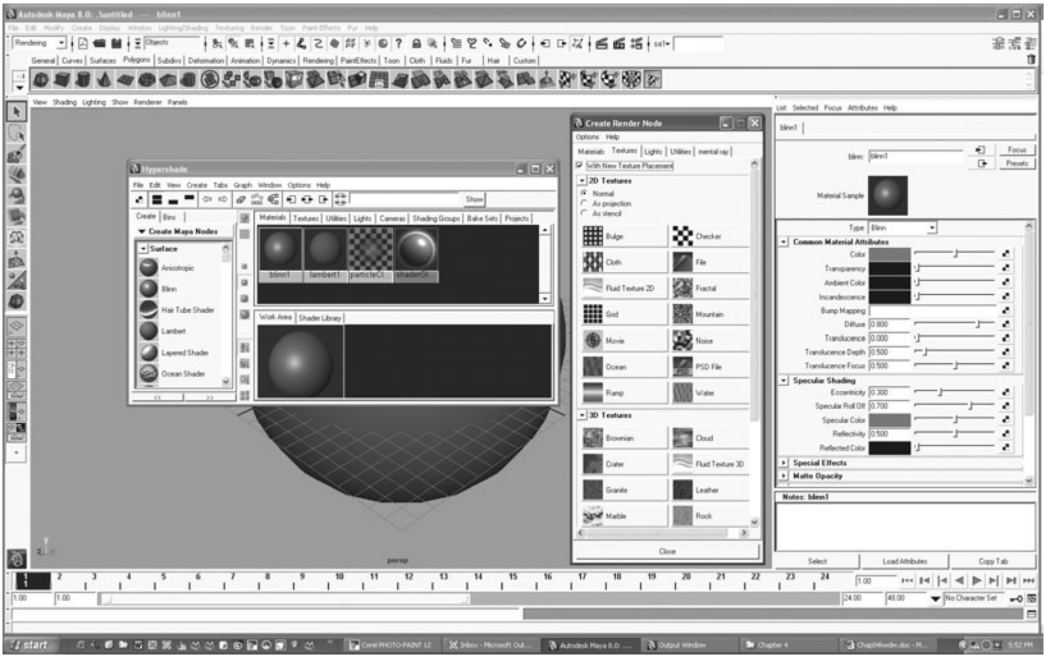


Figure 4.24
Launch the Create Render Node window.

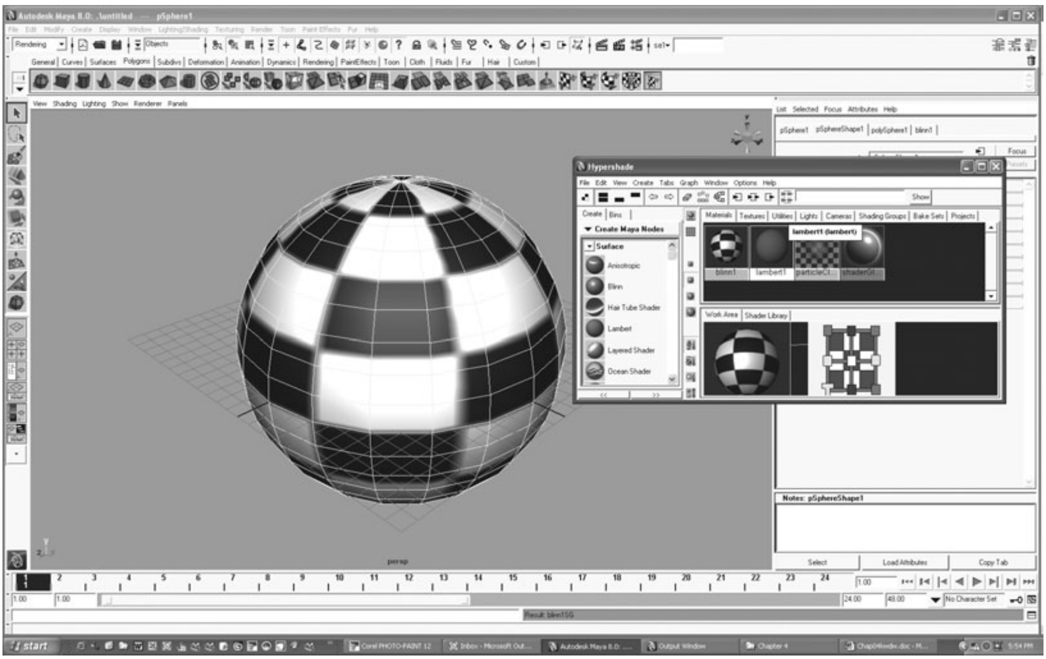


Figure 4.25
Apply a new material to the sphere.

Note

Marking menus are floating menus in Maya that are used for quick access to common functions. You access marking menus by selecting an item and then right-clicking it. When the marking menu appears, simply drag the cursor over the desired menu item and release the mouse button.

14. Click the red × button in the upper-right corner of the Hypershade window to close it.
15. Before you progress to the next section, you need to clear the scene. To do so, click the New Scene button in the Status line.

There you have it; you have created and textured your first model in Maya. That wasn't so hard, was it?

Menus

Although there are many buttons on the Maya interface, really understanding how to use the program will require an understanding of the layout of Maya's menus. This section shows how Maya's menus are arranged and how to use them. It also shows their main purposes. Because covering all the menu items in detail would be excessively boring, not all of them are covered here. Instead, you'll learn how to use Maya's menus by using them in animation projects throughout the book. In the meantime, take a moment to explore Maya's menus to get a sense of where everything is.

Note

If you are new to Maya, you may feel a little lost as you explore the menus because the menus contain many terms that may be unfamiliar. Don't worry too much about not knowing everything right now. Use this section as a reference. You will learn most of the terms associated with the menus through the projects in the book.

Menu Layout

Grouping menu items helps to define related functions. For this reason, each menu in Maya contains groups of commands separated by lines. In the File menu (see Figure 4.26), the first group contains two menu items: New Scene and Open Scene. Both are used to start a session in Maya, one by starting a new scene and the other by loading a saved scene.

Figure 4.26
The File menu is the first menu in the main menu.

Note

Notice the double line at the top of the File menu. This double line indicates that the menu is detachable. If you click this line, Maya copies the functions to a separate menu that you can place anywhere on the screen for easy access.

Menus in Maya have four columns:

- The first column, starting from the left, is the name of the menu item.
- The second column gives a keyboard shortcut for the menu item. The keyboard shortcut, or hotkey, enables you to access the function through the keyboard. You can see from the File menu that Ctrl+N is the keyboard shortcut for the New Scene command. This means that instead of clicking the New Scene button or selecting New Scene from the File menu, you also simply press Ctrl+N to create a new scene.
- The third column contains arrows. These indicate that if you select the menu item, a submenu will appear. For example, selecting the Projects menu item from the File menu reveals a submenu. You can then slide your cursor over the submenu and select one of the items it contains.

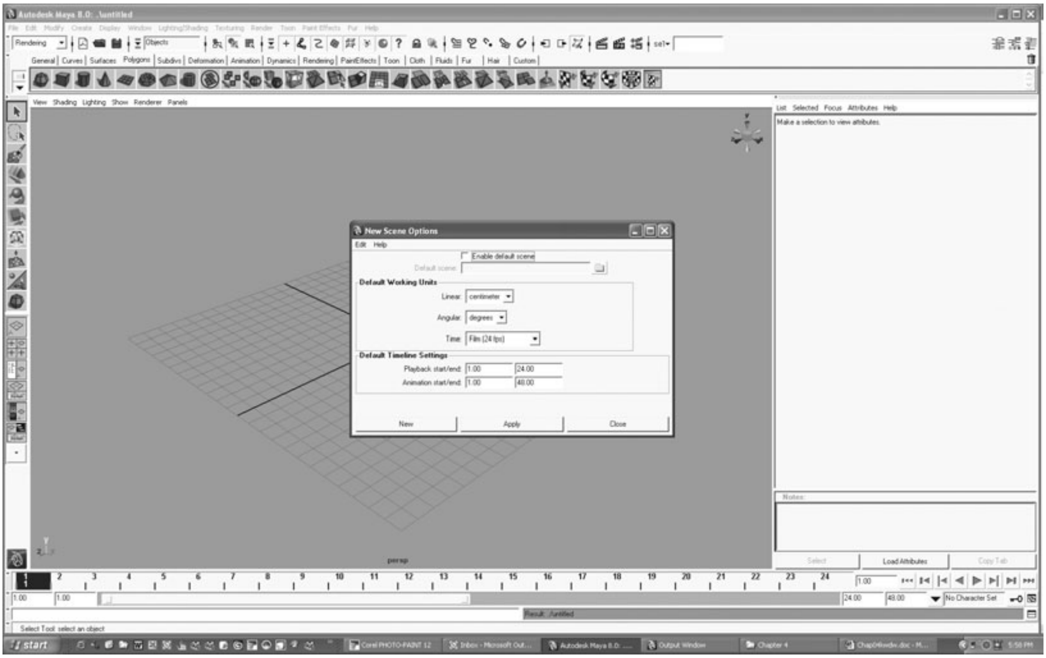


Figure 4.27
The New Scene dialog box.

- The fourth column indicates whether the menu item has a dialog box associated with it. If a box is present in this column, you can select it to launch a dialog box or panel, which you use to adjust settings for the menu function. For example, click the box icon next to the New Scene menu item; the dialog box shown in Figure 4.27 will appear. As shown, the settings in this dialog box relate to setting up new scenes.

Many dialog boxes, including the New Scene dialog box, are divided into sections. For example, the first section of the New Scene dialog box is for setting up custom new scenes. This is useful if you want to start every new scene with specific settings, such as a particular background or specialized geometry. The second section of the New Scene dialog box lets you customize your default working units. By default, centimeters are used, which is fine for small objects but becomes a problem if you are building, say, a diesel truck. Angles can be set to degrees or radians. Time relates to animation, and has a long list of options depending on the end target for your scene. Currently, Time is set for film animation. The third section of the New Scene dialog box lets you customize the length of the playback and the total number of frames in the animation. At the bottom of most dialog boxes, including the New Scene dialog box, are three buttons: Now, Apply, and Close. Clicking the Now button activates the function with the current settings and closes the dialog box. Clicking the Apply button activates the function with the current settings but leaves the dialog box open, useful if you need to apply function multiple times. The Close button closes the dialog box without activating the function. For now, close the New Scene dialog box and let's continue our tour of the menus.

Note

Dialog boxes are available via the menu, but not via hotkeys or onscreen buttons. Hotkeys and buttons are useful if you want to perform a function but don't need to change any settings first.

Common Main Menu Options

As shown in Figure 4.28, there are seven common menus in the main menu, plus a series of menus that vary depending on which option is selected in the drop-down list on the Status line. From left to right, these common menus are as follows:

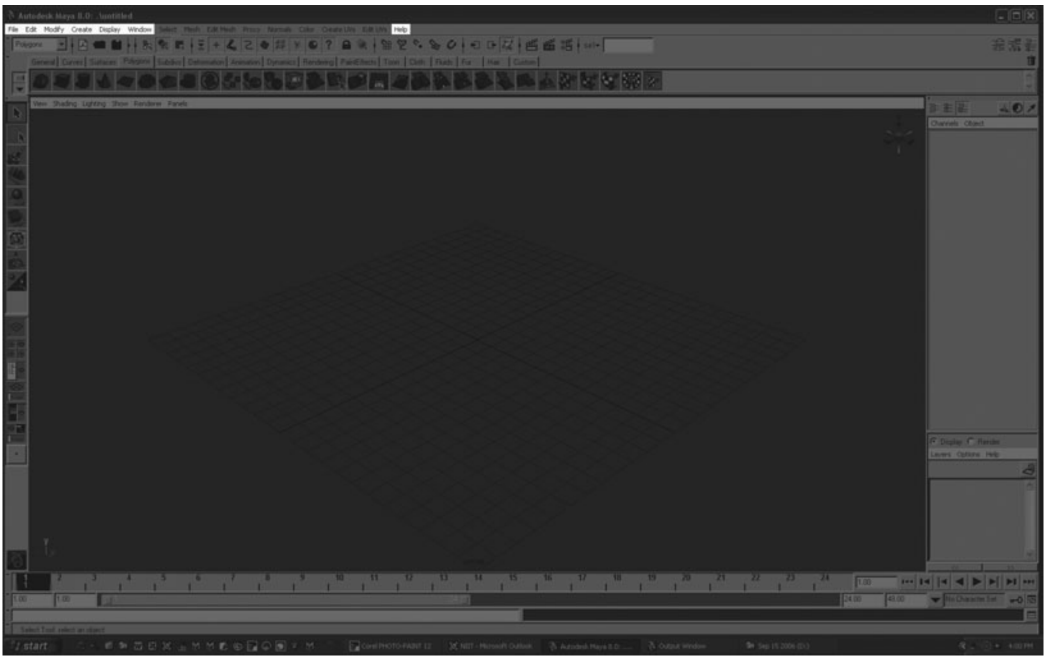


Figure 4.28
The common menus in the main menu.

Note

Some menu items that follow may not appear on the Maya PLE version that came with this book. This is because these menu items are only in the Maya Unlimited version of the software. They are included here to give you an idea of the full capabilities of the software.

- **File.** The File menu contains functions dealing with files, projects, importing, and exporting
- **Edit.** This menu contains functions dealing with basic editing, selecting, grouping, and parenting operations.
- **Modify.** The Modify menu contains functions dealing with modifying geometry or other scene elements.
- **Create.** The Create menu contains functions dealing with the creation of scene elements like geometry, lights, curves, and text.
- **Display.** This menu contains functions dealing with onscreen display elements including objects, components, and interfaces.
- **Window.** The Window menu contains functions dealing with the many editors and tools in Maya.
- **Help.** This menu contains functions dealing with online help, tutorials, references, and product information.

The Animation Menu Options

When Animation is selected from the drop-down list on the Status line, the main menu features six Animation menu items (see Figure 4.29). From left to right, they are as follows:

- **Animate.** This menu contains functions dealing with keys, clips, and paths.
- **Deform.** The Deform menu contains functions dealing with high-level tools used to manipulate low-level components.
- **Skeleton.** This menu contains functions dealing with the creation and editing of joints as well as animation helpers.
- **Skin.** The Skin menu contains functions dealing with attaching geometry to skeletal systems.

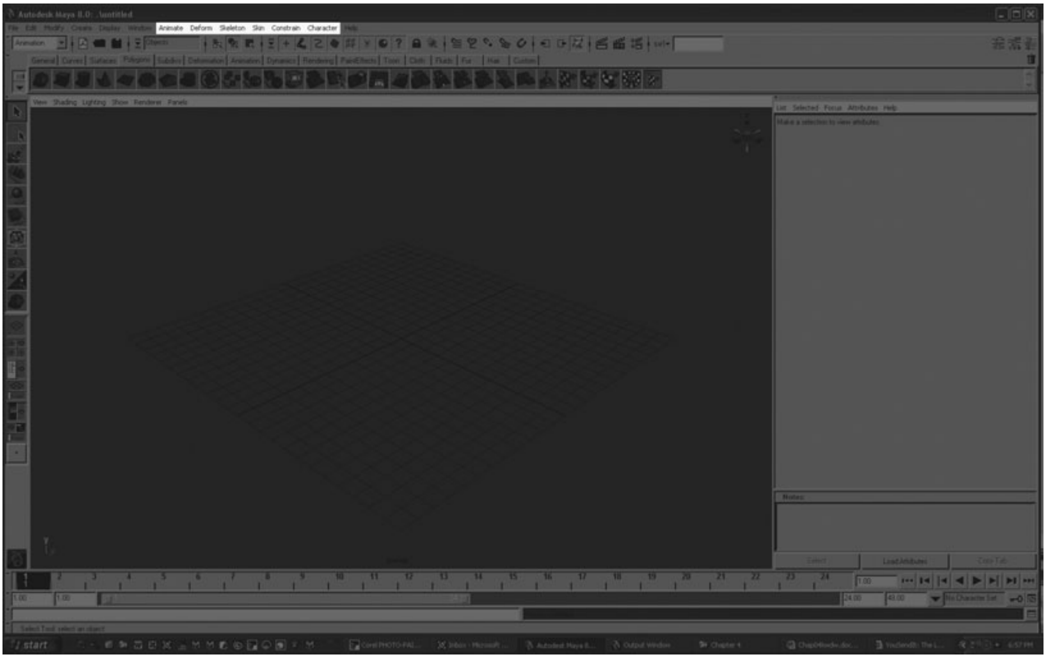


Figure 4.29
The Animation menu set includes six menu items.

- **Constrain.** This menu contains functions dealing with setting limits or constraints on animated objects.
- **Character.** This menu contains functions dealing with the creation and editing of characters.

The Polygons Menu Options

When Polygons is selected from the drop-down list on the Status line, the main menu features eight modeling menu items (see Figure 4.30). From left to right, they are as follows:

- **Select.** The Select menu deals with selection options for polygon objects and components.
- **Mesh.** The Mesh menu deals with modifying polygon objects
- **Edit Mesh.** The Edit Mesh menu deals with editing polygon object components.

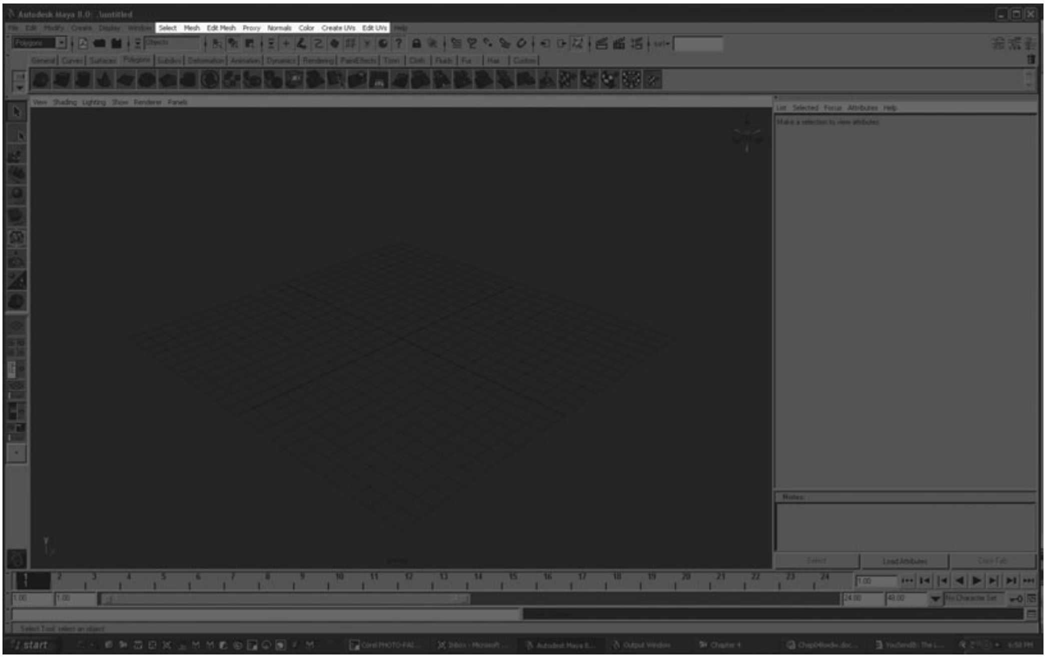


Figure 4.30
The Polygons menu set includes eight menu items.

- **Proxy.** This menu is for creating and editing a polygon proxy for subdivision surface models.
- **Normals.** This menu is used for editing polygon normals.
- **Color.** This menu is used to edit vertex color on polygon objects.
- **Create UVs.** The Create UVs menu contains functions for applying and editing UVs on polygon objects.
- **Edit UVs.** The Edit UVs menu contains functions for editing UVs once they have been placed on an object.

The Surfaces Menu Options

Surfaces menu options control the building of objects using NURBS and Subdivision Surfaces. There are four main menu items in the Surfaces menu, as shown in Figure 4.31. They are from left to right as follows:

- **Edit Curves** This menu contains functions dealing with editing and modifying curves

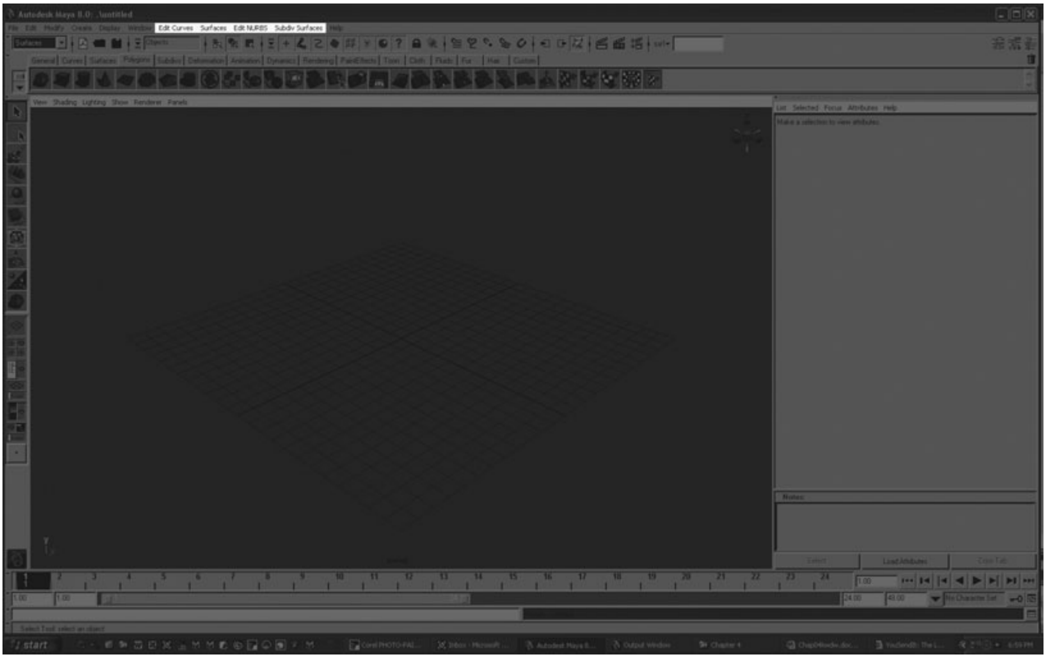


Figure 4.31
The Surfaces menu set includes four menu items.

- **Surfaces.** The Surfaces menu contains functions dealing with creating and editing surfaces.
- **Edit NURBS.** This menu contains functions dealing with NURBS modeling and NURBS editing.
- **Subdiv Surfaces.** This menu deals with building and editing subdivision surfaces.

The Dynamics Menu Options

When Dynamics is selected from the drop-down list on the Status line, the main menu features seven Dynamics menu items (see Figure 4.32). From left to right, they are as follows:

- **Particles.** This menu contains functions dealing with the creation and editing of particles
- **Fluid Effects.** The Fluid Effects menu contains functions for creating and editing fluid effects.

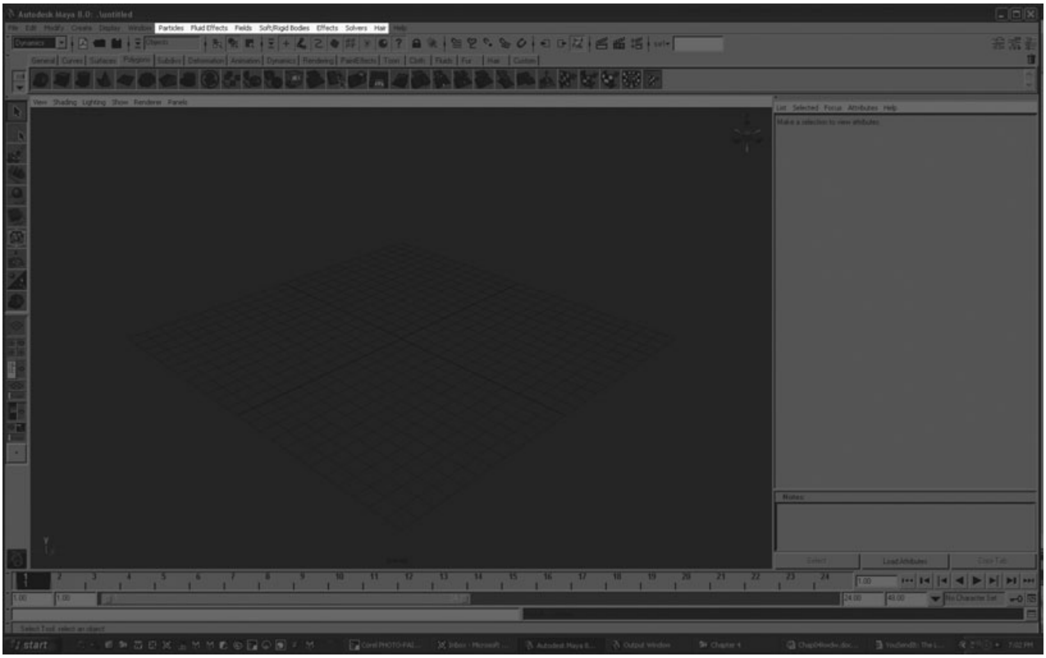


Figure 4.32
The Dynamics menu set includes seven menu items.

- **Fields.** The Fields menu contains functions dealing with the creation of fields.
- **Soft/Rigid Bodies.** This menu contains functions dealing with the creation and editing of soft and rigid body objects.
- **Effects.** The Effects menu contains functions dealing with the creation of specialized effects such as fire or lightning.
- **Solvers.** This menu contains functions dealing with a collection of specialized solutions for dynamics animation.
- **Hair.** The Hair menu contains functions for creating and editing Hair effects.

The Rendering Menu Options

When Rendering is selected from the drop-down list on the Status line, the main menu features six Rendering menu items (see Figure 4.33). From left to right, they are as follows:

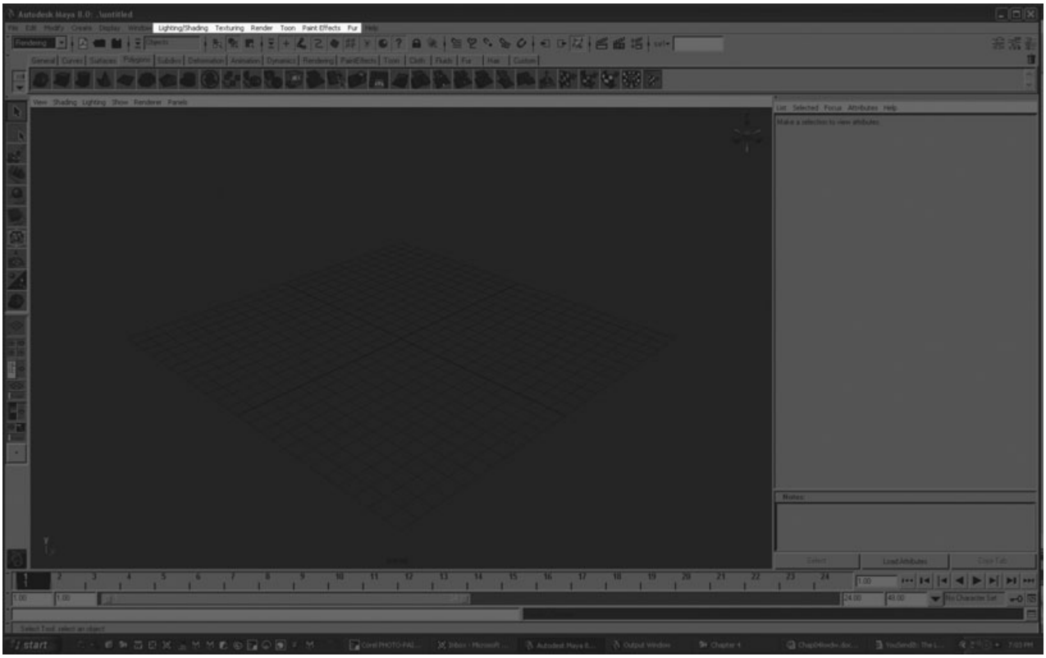


Figure 4.33
The Rendering menu set includes six menu items.

- **Lighting/Shading.** This menu contains functions dealing with editing materials and lights.
- **Texturing.** The Texturing menu contains functions dealing with 3D paint, PSD, texture referencing, and NURBS texture application.
- **Render.** This menu contains functions dealing with the rendering of scenes.
- **Toon.** The Toon menu contains functions dealing with Maya's Toon Shader.
- **Paint Effects.** This menu contains functions dealing with creating paint effects.
- **Fur.** The Fur menu contains functions for creating and editing fur

The Cloth Menu Options

The cloth menu set is used for controlling cloth effects in Maya. Cloth is an advanced feature only available in Maya Unlimited. As shown in Figure 4.34, it contains three menu items. Listed from left to right, they are as follows:

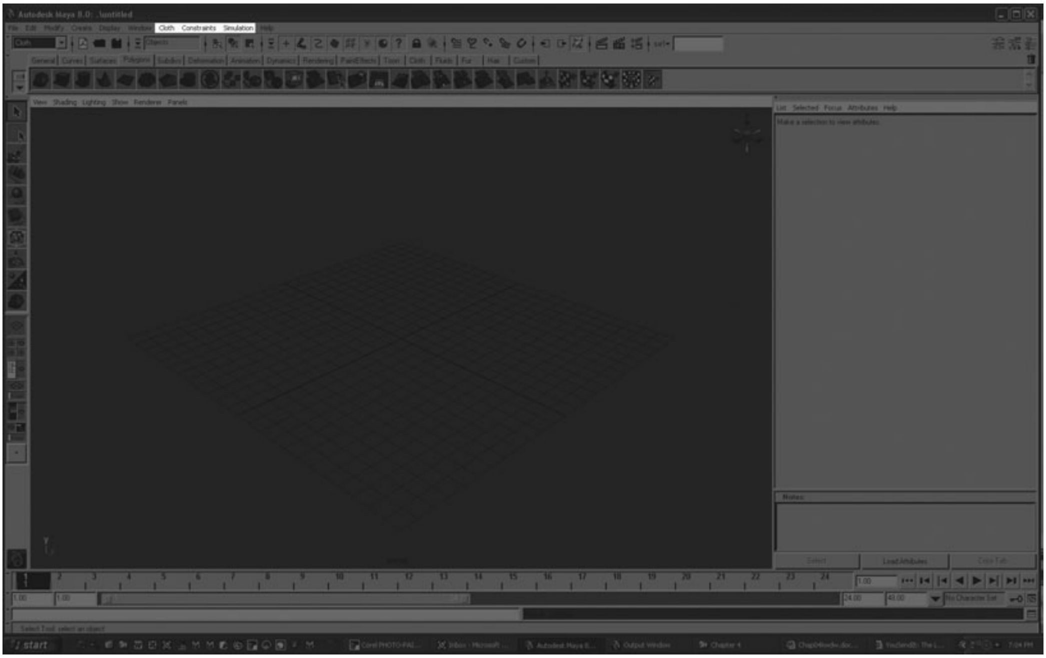


Figure 4.34
The Rendering menu set includes six menu items.

- **Cloth.** The Cloth menu contains functions for creating and defining cloth objects.
- **Constraints.** The Constraints menu contains menu items used to constrain the movement and reaction of cloth objects.
- **Simulation.** The simulation menu contains tools for simulating the effects of cloth objects.

Polygon Modeling Tools

In simple terms, building 3D game characters is the process of creating a virtual 3D representation of a character on the computer that can later be exported into a game engine. This section looks at creating simple objects using primitives. *Primitives* are basic polygon objects—that is, geometric shapes like cubes, spheres, cones, and so on. Modeling using primitives is very common; even complex models have their roots in basic shapes.

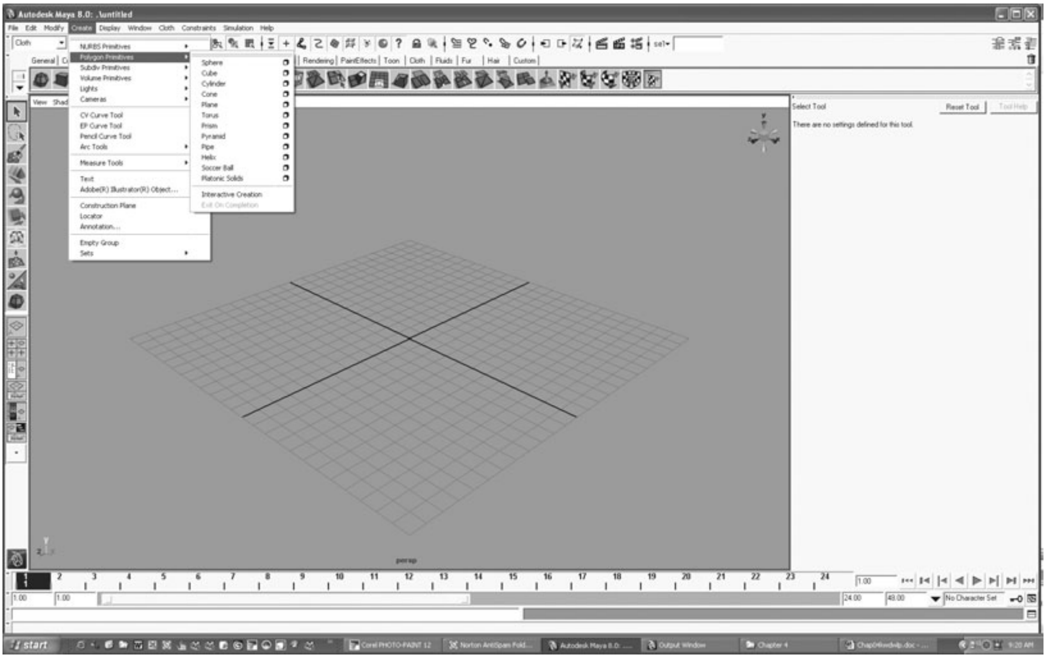


Figure 4.35
The Polygon Primitives command is located in the Create menu.

Maya has many tools that are useful for building models with polygons. Understanding these tools will help you to master the techniques of building models with them. Some of these tools were used in Chapter 2, "Developing Character Ideas," to create the ball. Now is a good time to explain them in greater detail. First, take a look at creating polygon primitives; then examine how these primitive objects can be modified.

Creating Polygon Primitives

You can find the Polygon Primitive creation tools on the Create menu, as shown in Figure 4.35.

Note

At the bottom of the Polygon Primitives submenu is an Interactive Creation option. This was added in Maya 8 to support the interactive creation of objects in the workspace. The Interactive Creation feature works best for objects that are not a specific size and are not in a specific location. If you want to create objects that are defined exactly in size and location, you should turn this option off. If you want to sculpt your objects in a more freehand, visual way, you should select this menu item. Note, however, that the freehand modeling technique is a more advanced approach, for modelers who are already familiar with 3D modeling and use a more organic approach to constructing models. For the purposes of this book, we will leave these features unchecked to help with the creation of models that are scaled and placed appropriately in the scene. If you are a more advanced modeler, you can use the freehand tools to create the projects in the book; just be aware that you may need to move and scale the objects after creation to get the right size and placement.

The twelve polygon primitives supported by Maya and available from the Create menu are as follows. Each option offers a dialog box for setting the creation options such as the size, shape, and number of polygons in the object.

- **Sphere.** Figure 4.36 shows the Polygon Sphere Options dialog box. The first option, Radius, defines the size of the sphere; you can set the number in the box by using the slider or by typing it directly in the box. The next two options, which also use sliders, set the number of polygons in the sphere. The Axis option buttons establish the orientation of the sphere. The Texture Mapping options control how the UVs are placed on an object. *UVs* are points that control the placement of textures. If the UV checkbox is checked, UVs will be placed on an object. If it is unchecked, the object will not have any UV information. Without UVs, textures can't be placed on an object. The radio buttons below the UV checkbox are options for the placement of the UVs at the poles of the sphere.

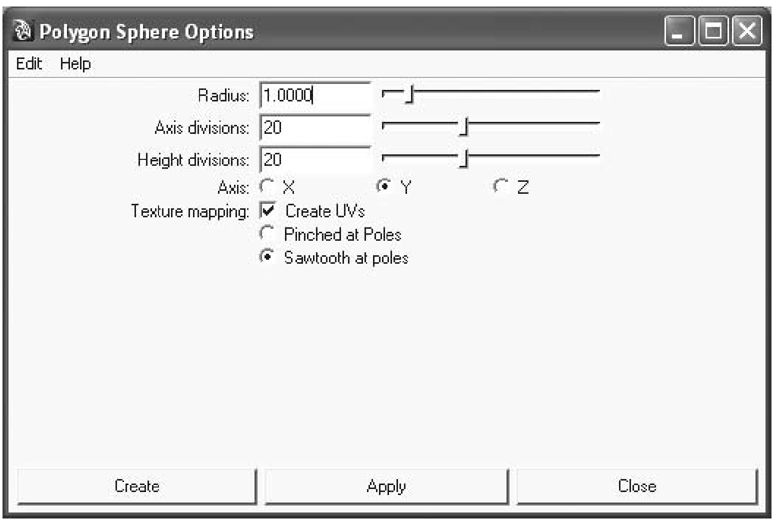


Figure 4.36
The Polygon Sphere Options dialog box.

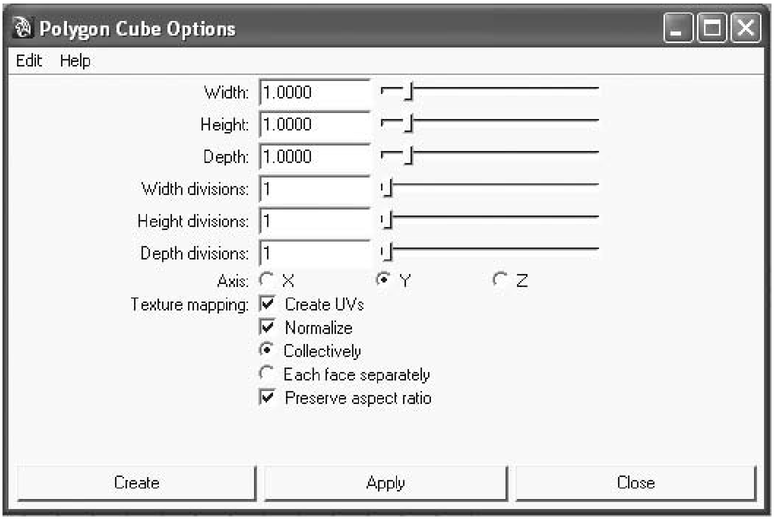


Figure 4.37
The Polygon Cube Options dialog box.

Note

Maya remembers the options for dialog boxes. If you want to make several primitives with the same options, you don't need to go back into the primitive's dialog box. Just selecting the primitive on the menu or on the Shelf will create a new primitive with the same options as the last one created. This feature in Maya is called *persistence*.

- **Cube.** The Polygon Cube Options dialog box is shown in Figure 4.37. The first three fields in this dialog box control the width, height, and depth of the cube, and the next three fields control the number of polygons in each. The Axis option buttons control the axis of the cube.

Common Options for Polygon Primitives

Each polygon primitive will have UV options. These options affect how UVs are organized on the object. The following are common options for polygon primitives:

- **Create UVs.** If this box is checked, UVs will be created with the object.
- **Normalize.** If this box is checked, it forces all the UVs to be inside the 0 to 1 texture area. This is the optimum texture area for objects and should be left on in most instances.
- **Collective.** Collective maps the texture over the entire surface of the object, as shown in Figure 4.38.
- **Each Face Separately.** This option maps the texture to each plane as shown in Figure 4.39.
- **Preserve Aspect Ratio.** This option does not distort the UVs when placed on the texture plane. When this option is turned off, the UVs are stretched to fit the texture area.

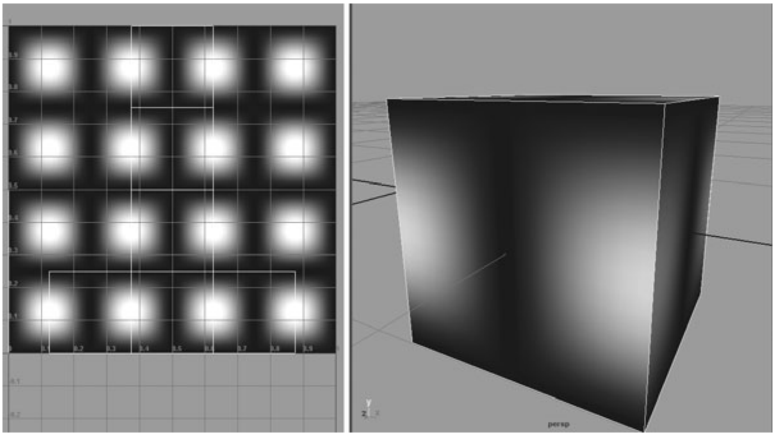


Figure 4.38
Collective maps the UVs across the texture.

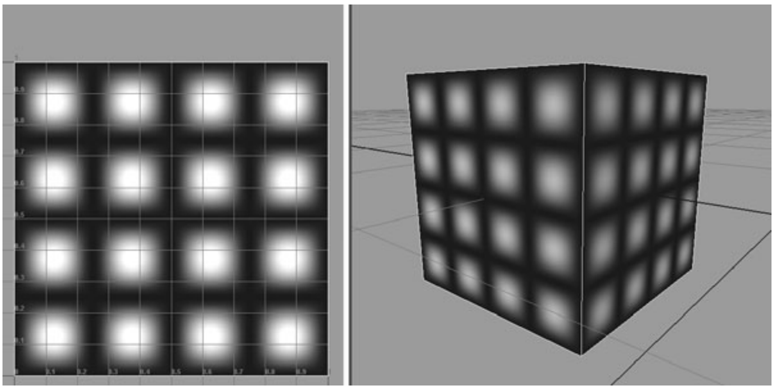


Figure 4.39
Each Face Separately maps the texture to each UV.

- **Cylinder.** Figure 4.40 shows the Polygon Cylinder Options dialog box. Like the sphere, the cylinder has a radius, but it also has a height—hence the first two fields in the dialog box, which control the cylinder's radius and height. The next three fields control the number of polygons around the axis, height, and cap of the cylinder, and the Axis option buttons control the axis. Like the cube, the Polygon Cylinder Options dialog box also has options for applying the UVs.
- **Cone.** Figure 4.41 shows the Polygon Cone Options dialog box. The options for the cone are very similar to the options for the cylinder.

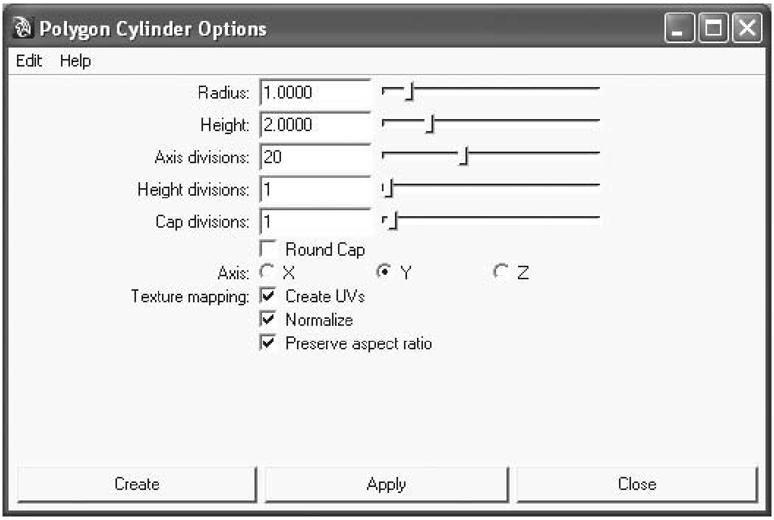


Figure 4.40
The Polygon Cylinder Options dialog box.

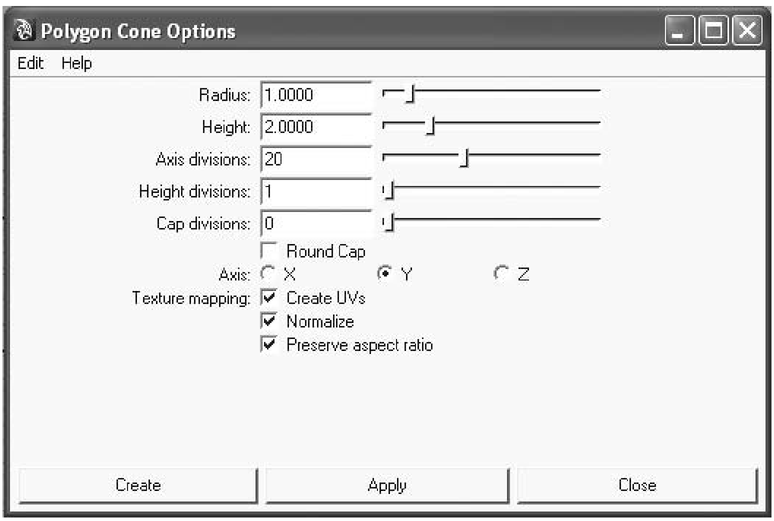


Figure 4.41
The Polygon Cone Options dialog box.

- **Plane.** Figure 4.42 shows the Polygon Plane Options dialog box. A *plane* is a flat polygonal object. It has only two directions for size and two directions for polygons, but has an axis like other objects. The default texture mode for a plane is Stretch is Preserve Aspect Ratio. Figure 4.43 shows two planes. The one on the right has Preserve Aspect Ratio selected and the one on the left does not. For most objects, it is better to not have Preserve Aspect Ratio checked.

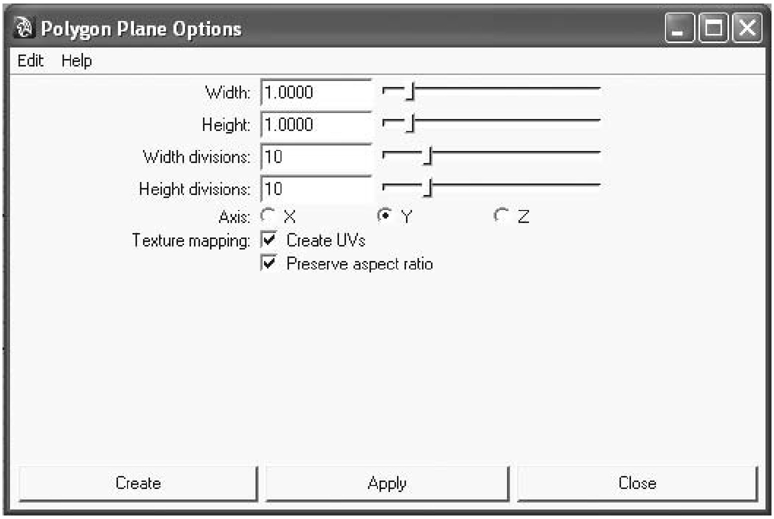


Figure 4.42
The Polygon Plane Options dialog box.

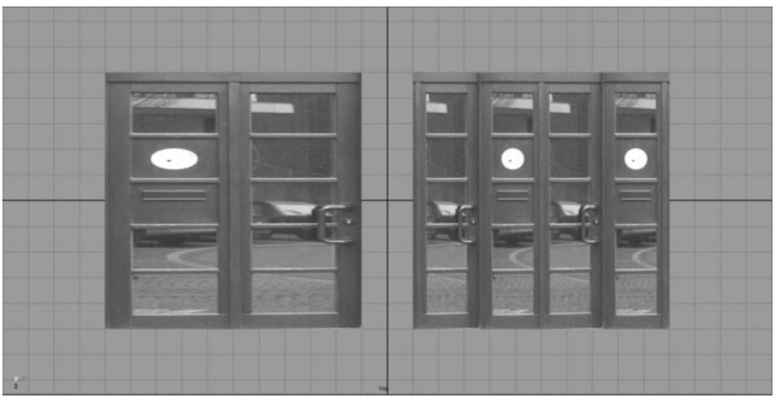


Figure 4.43
In most cases it is preferable to turn Preserve Aspect Ratio off when creating polygon planes.

- **Torus.** Figure 4.44 shows the Polygon Torus Options dialog box. This dialog box is similar to the other polygon dialog boxes. The first three fields deal with the size of the object, the next two fields deal with the number of polygons in the object, the option buttons deal with the object's axis, and the other option relates to texture.
- **Prism.** Figure 4.45 shows the Polygon Prism Options dialog box. This dialog box follows the same pattern as the others; indeed, it has the exact same

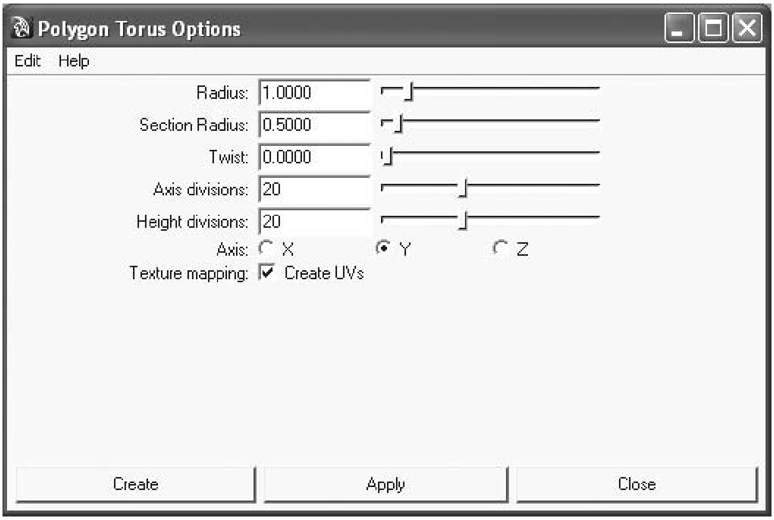


Figure 4.44
The Polygon Torus Options dialog box.

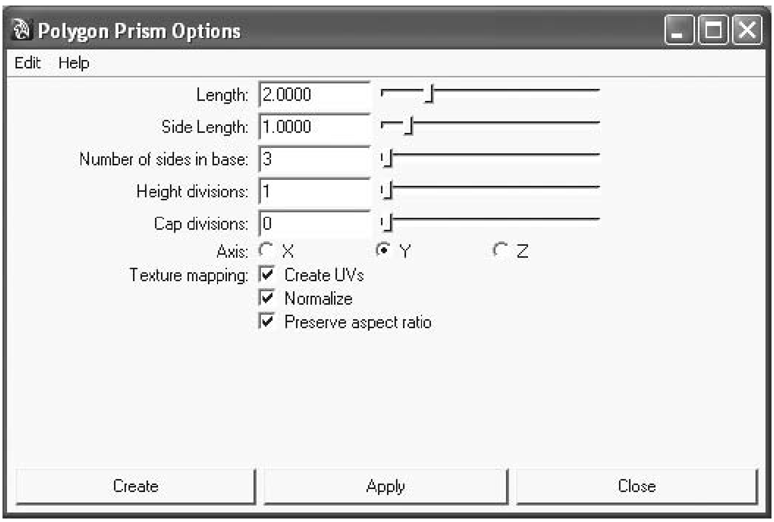


Figure 4.45
The Polygon Prism Options dialog box.

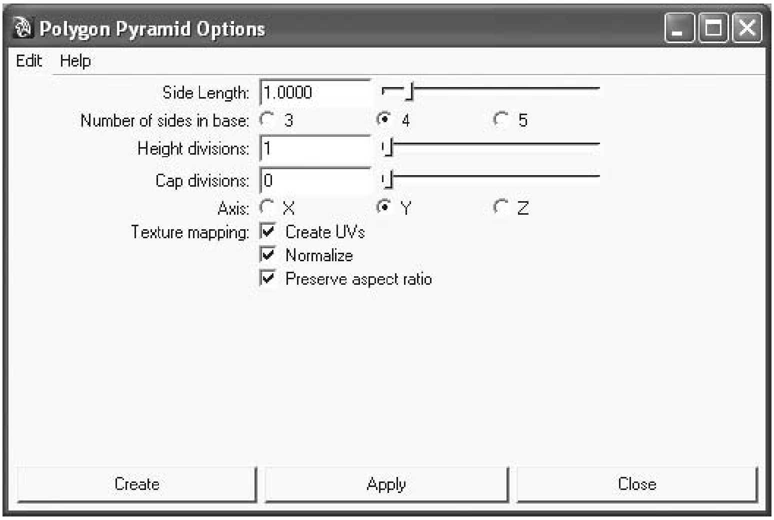


Figure 4.46
The Polygon Pyramid Options dialog box.

settings as the Polygon Cylinder Options dialog box. The difference is that edges on a prism are sharp rather than soft, as on a cylinder.

- **Pyramid.** Figure 4.46 shows the Polygon Pyramid Options dialog box. This dialog box is a little different from the others; it has only one size field because all the sides are the same length. The pyramid can have a three-, four- or five-sided base, which you select using the dialog box's option buttons. The Subdivisions, Axis, and Texture options are like similarly named options in other dialog boxes.
- **Pipe.** Figure 4.47 shows the Polygon Pipe Options dialog box. This dialog box is similar to the Polygon Cylinder Options dialog box with one major exception: The Polygon Pipe Options dialog box contains a Thickness field, which specifies the distance from the inner cylinder to the outer cylinder.
- **Helix.** Figure 4.48 shows the Polygon Helix Options dialog box. A helix, which looks like the coil of a spring, is more complex than other polygon primitives. Hence, the Polygon Helix Options box contains a few extra settings. The first field, Coils, specifies the number of coils in the helix. The next two fields set the height and width of the coils. The last size-related field establishes the radius of the coil. As with the other option boxes, the size

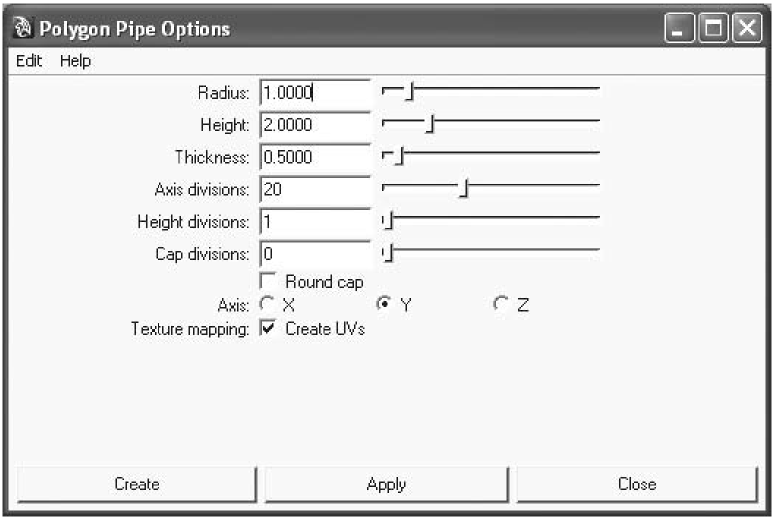


Figure 4.47
The Polygon Pipe Options dialog box.

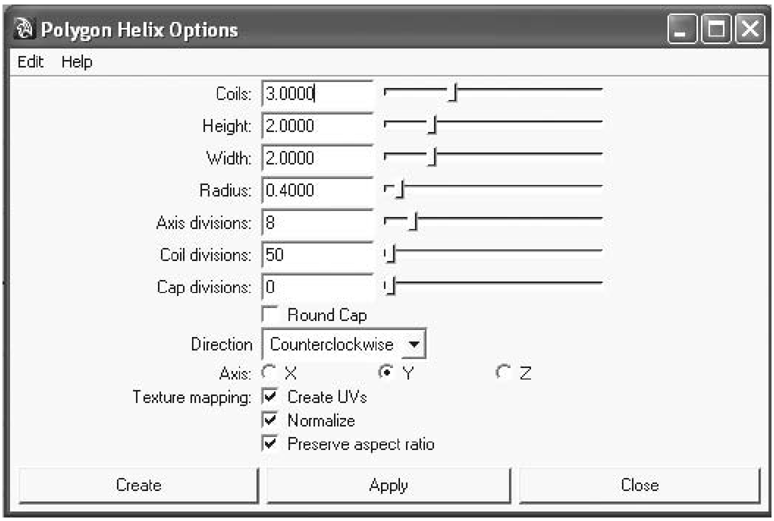


Figure 4.48
The Polygon Helix Options dialog box.

settings are followed by Subdivision options. There are three of them—Subdivisions Around Axis, Subdivisions per Coil, and Subdivisions on Caps. The next setting is for the spiral direction, and can be either counterclockwise or clockwise. The Axis and Texture options, which are similar to the ones in other polygon primitive dialog boxes, follow.

- **Soccer ball.** Figure 4.49 shows the Polygon Misc Options dialog box, used to set the options for the Soccer Ball primitive. The first thing that you might notice about this dialog box is that there are no subdivision options; That's because the object already has a predefined subdivision. If you try to change the size options, you will notice that the two are tied together.

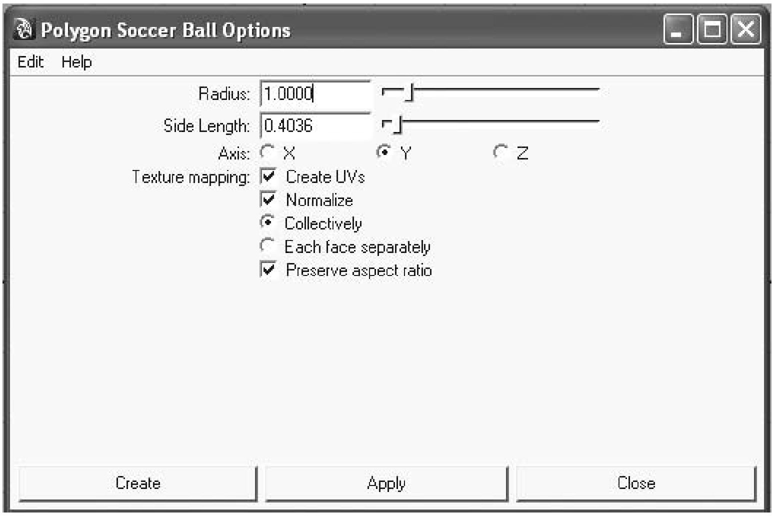


Figure 4.49
The Polygon Misc Options dialog box.

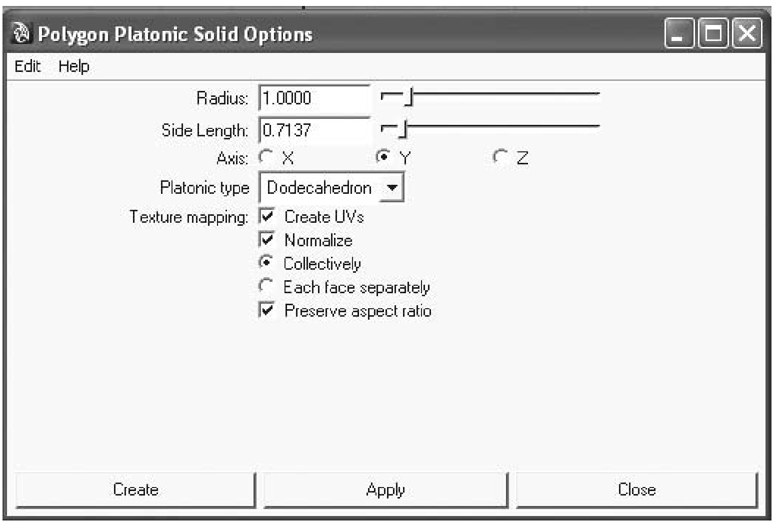


Figure 4.50
The Polygon Platonic Solid Options dialog box.

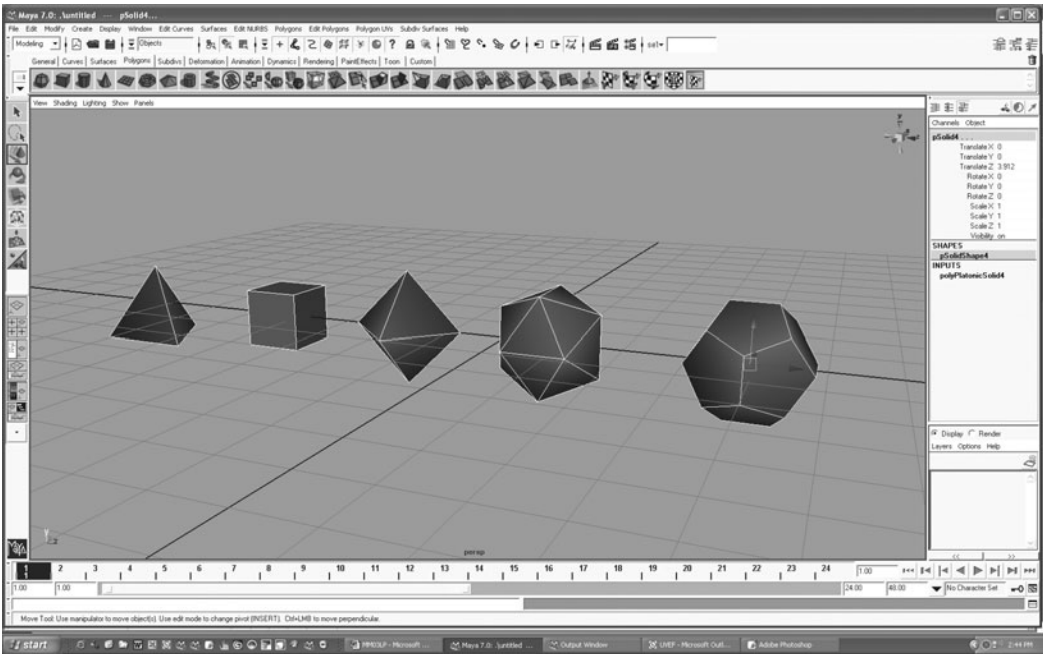


Figure 4.51
There are only five platonic solids.

If you increase or decrease one, the other one will automatically change to match. The Axis and Texture options are similar to those options in other dialog boxes.

- **Platonic solids.** Figure 4.50 shows the Platonic Solid Options dialog box. Platonic solids are geometric objects whose edges, angles, and faces are all the same. There are only five platonic solids: tetrahedron, hexahedron (cube), octahedron, dodecahedron, and icosahedron, all of which are shown in Figure 4.51. The Polygon Platonic Solid Option dialog box is similar to the Polygon Misc Options dialog box except that the dialog box for platonic solids features a drop-down list below the size fields that enables you to choose which platonic solid to create. (Note that cube is not included because it is already one of the polygon primitives in Maya.)

Modeling Polygon Primitives

Maya has a separate menu set for modeling polygons as discussed above. Change the menu set to Polygons. You don't need to know every function in these menus for now, but there are several that are very important to understand.

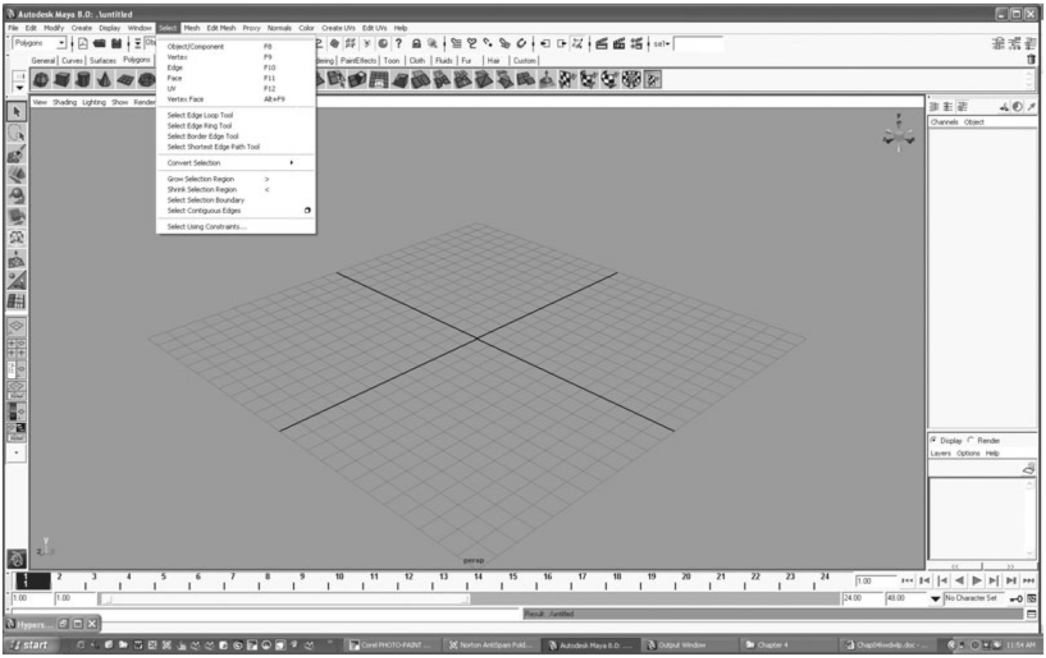


Figure 4.52
The Select menu.

Using the Select Menu

The Select menu reveals several options for selecting component types of an object (see Figure 4.52). With the selection options you can change the type of component you wish to select. You can also select groups of components like edge loops or border edges.

Convert Selection is very useful when a selection does not correspond to a function. For example, UVs often need to be rotated on specific faces. It is easy to select the faces but hard to select the specific UVs for those faces. Selecting the faces and then using Selection to change the selection to UVs is a simple way to select only the UVs you want to rotate.

Using the Mesh Menu

The Mesh menu, shown in Figure 4.53, includes several functions for modeling a polygon primitive:

- **Combine.** Combine is a commonly used function that combines all selected objects into one object. Say, for example, that you are creating

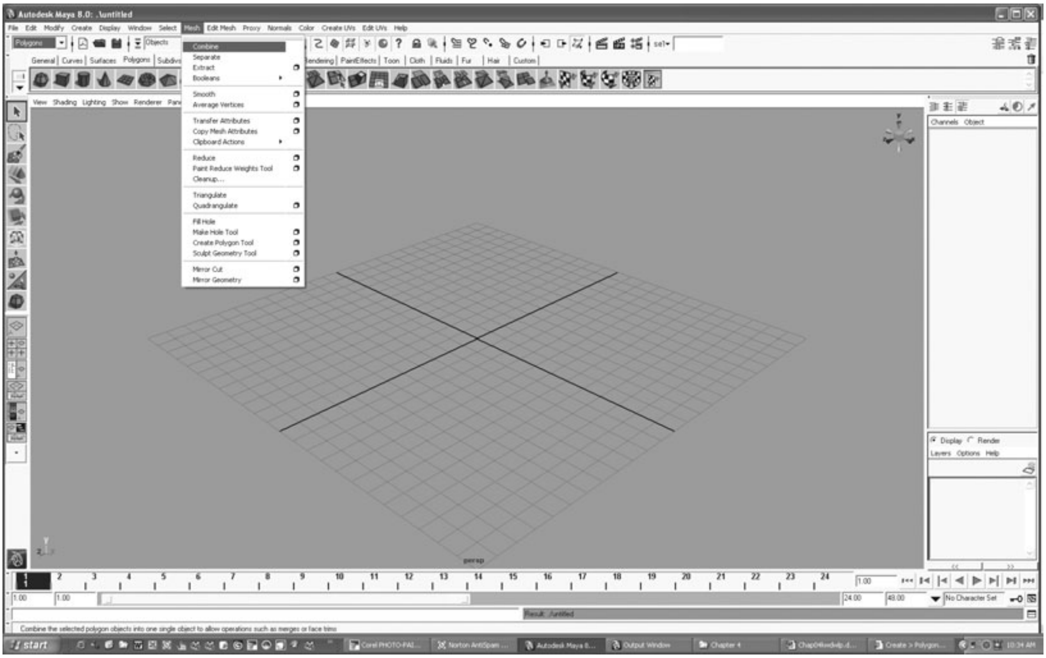


Figure 4.53
The Mesh menu.

a tree, whose leaves are made up of many small polygons. You can combine all the leaves into one object by using the Combine function.

- **Separate.** Separate is used to separate unmerged polygons from each other. For example, in Chapter 6, "Character Modeling II," the Combine function is used to create a single object from several branch and leaf objects. The Separate function could be used to separate the original objects from the combined object. This is useful if modifications are needed to in the original objects that comprise the combined object.

- **Extract.** The Extract function detaches a selected set of polygons from a model, creating a separate object. It is used for separating model parts such as a door on a car or a window in a building.

- **Smooth.** The Smooth function adds polygons to an object to reduce the sharp edges of a model. The function averages the angles of the polygons to make the surface of a model more rounded. Modelers often use Smooth because making a simple object is faster than a making a complex one. After the basic object is created using just a few polygons, the artist can then use the Smooth function to round the rough-faceted surface.

- **Average Vertices.** The Average Vertices function smoothes an object or selection by moving the vertices rather than creating new geometry. This function is useful when the model needs to retain a set number of polygons but needs to be less irregular. The down side of the Arrange Vertices function is that because the vertices move, modeling problems may result.
- **Cleanup.** Cleanup is used to find and correct problems in a polygonal model. It has a number of options that let the artist define the parameters of a search to detect specific problems. It is similar to a spelling and grammar check in a word processor.
- **Sculpt Geometry Tool.** This function has many uses, but its main use is to interactively move polygons, similar to sculpting clay.
- **Mirror Geometry.** A commonly used function in Maya is Mirror Geometry. This function mirrors an object across a plane. It is frequently used for symmetrical models, enabling the modeler to focus on building only half the model and then mirroring the model to complete it. A good example of when using the Mirror Geometry function would be useful is when modeling a car because the modeler needs to build only half of the car. Then, with the Mirror Geometry function, the other half can be added at the end of the modeling process.

Using the Edit Mesh Menu

The Edit Mesh menu, shown in Figure 4.54, has many useful tools and functions for modifying elements of a polygon object. The functions in the Edit Mesh menu differ from those in the Mesh menu in that Mesh menu items are used on polygon objects, where Edit Mesh menu items are for modifying *components* of an object. When using functions in the Edit Mesh menu, the selected item should generally be a component of a polygon rather than an object.

- **Extrude.** Extrude is a commonly used tool. It allows the artist to select a number of individual components and extrude them. Extrude is commonly used to add features to primitive models—for example, adding arms and legs to a character or building walls from a floor plan of a room. The Extrude feature works with faces, edges and vertices.
- **Split Polygon.** The Split Polygon function is used to divide individual polygons when specific geometry is needed to complete a model. The tool lets the artist draw in edges on a polygon.

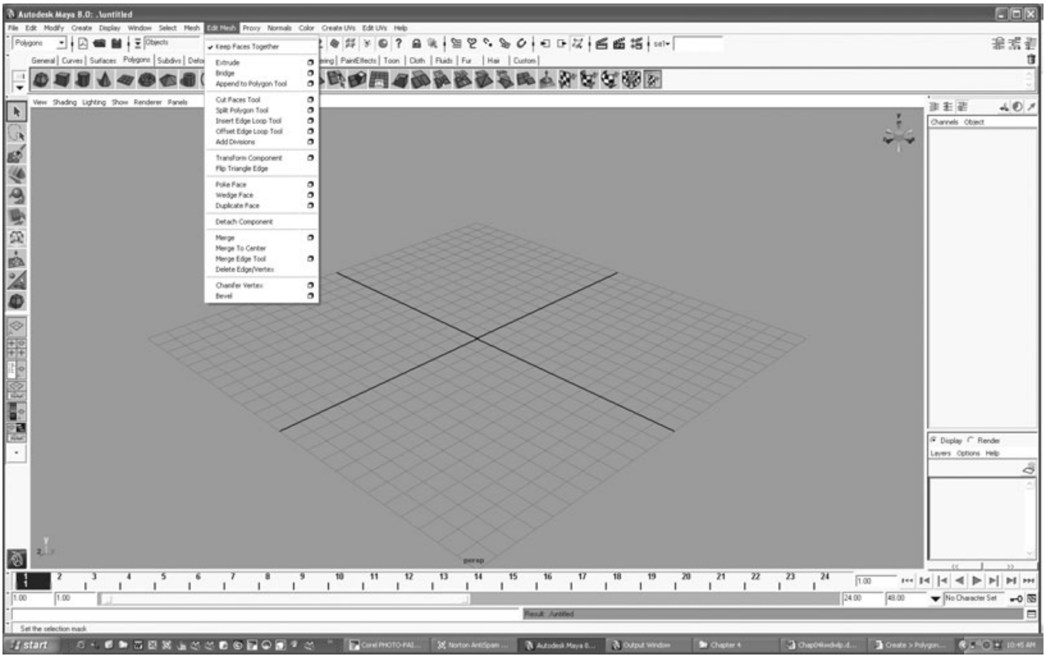


Figure 4.54
The Edit Mesh menu.

- **Insert Edge Loop Tool.** Edge loops are used to add a single plane edge to an object. Say fore example, you want to add some polygons to an arm so you can increase the detail. You can use the Insert Edge Loop to insert an edge loop dividing the polygons as shown in Figure 4.55.
- **Offset Edge Loop Tool.** The Offset Edge Loop tool places a new edge on either side of an existing edge along the entire length of that edge as shown in Figure 4.56.
- **Add Divisions.** The Add Divisions function is used to increase the density of the polygon mesh in selected areas of a model. The Subdivide function splits selected faces evenly, and is used when a denser mesh is needed in specific locations on a model.
- **Duplicate Face.** The Duplicate Face function makes a copy of a selected set of polygon faces. It is often used to create copies of repetitive polygons—for example, when building a fence or creating wheels for a car.

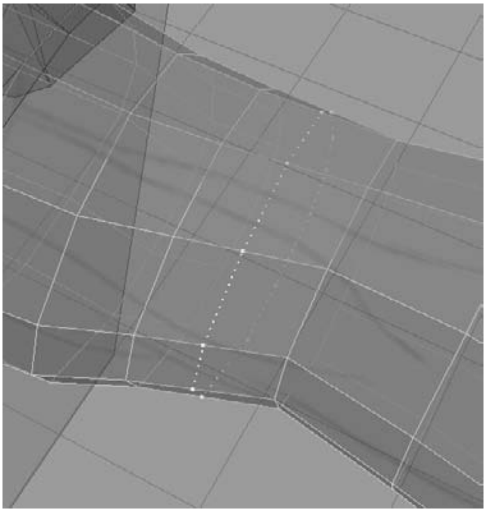


Figure 4.55
Use edge loops to divide polygons along a plane.

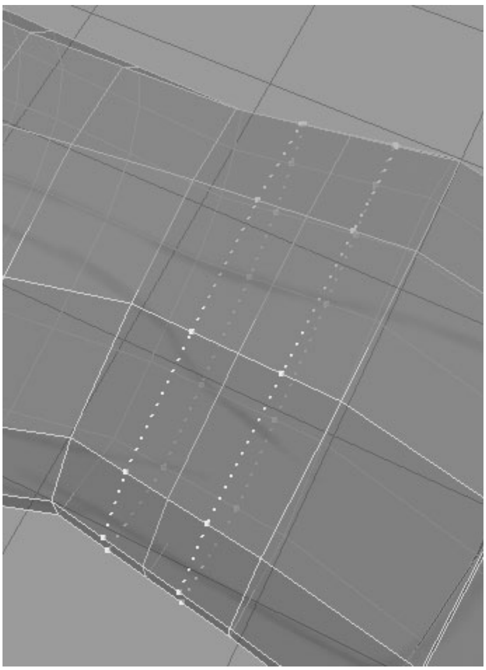


Figure 4.56
Offset Edge Loop adds new edges along both sides of an existing edge.

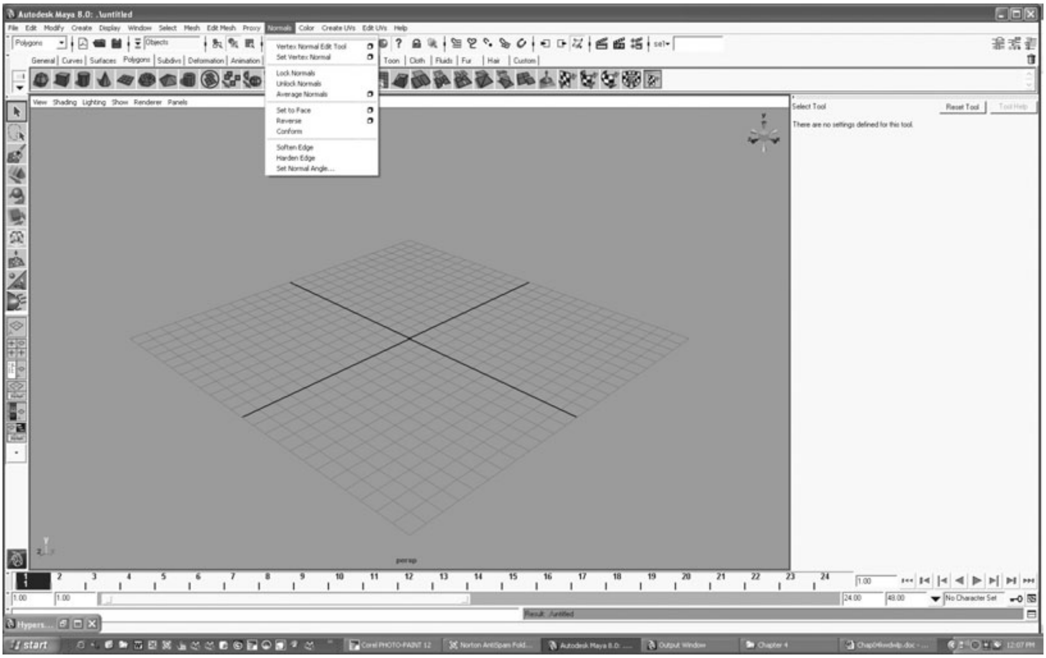


Figure 4.57
Choose Normals to see your options for adjusting the direction of a normal on a polygon.

- **Merge.** The Merge functions are used to combine two or more components into a single one. Merge is often used to join two parts of an object or to reduce the number of polygons in an area.

A polygon normal indicates which direction a polygon surface is facing. When a texture is applied to a polygon, the texture will appear correctly if the polygon normal is facing the viewer. If the polygon normal is facing away form the viewer, the texture will be reversed. Choosing the Normals menu reveals a menu of functions for adjusting the direction of the normal on a polygon (see Figure 4.57).

Building a Simple Character Model

In this section, you will use the knowledge you just gained to create a simple geometric character from a cylinder and a sphere. To begin, select New from the File Menu to bring up a new scene. Then do the following:

1. Create a cylinder with a radius of 4 units and a height of 12 units. Change the Subdivisions Around Axis setting to 12, the Subdivisions Along Height

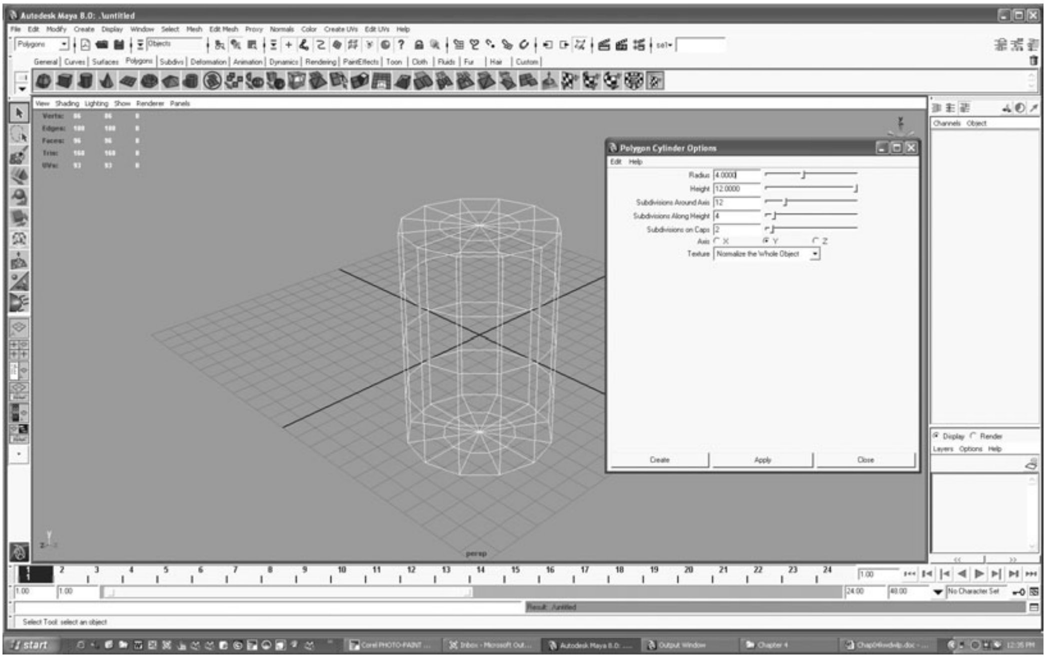


Figure 4.58
Create a cylinder for the character's torso.

setting to 4 and the Subdivisions on Cap setting to 2, as shown in Figure 4.58. The cylinder will be used to form the character's torso.

2. Use the Scale tool to scale in the cylinder in the z axis as shown in Figure 4.59.

3. In Front view, right-click the model to open the Marking menu, a floating menu with many common Maya functions. In the Marking menu, change the selection mode to vertex. (You'll need to hold the right mouse button down until the cursor selects the Vertex option.) Next, use the Scale tool to taper the character's chest and hips as shown in Figure 4.60.

4. Select the vertices on the outer shoulders and use the Move tool to pull them down as shown in Figure 4.61.

5. In Side view, expand the character's chest by using the Scale tool in the z axis as shown in Figure 4.62.

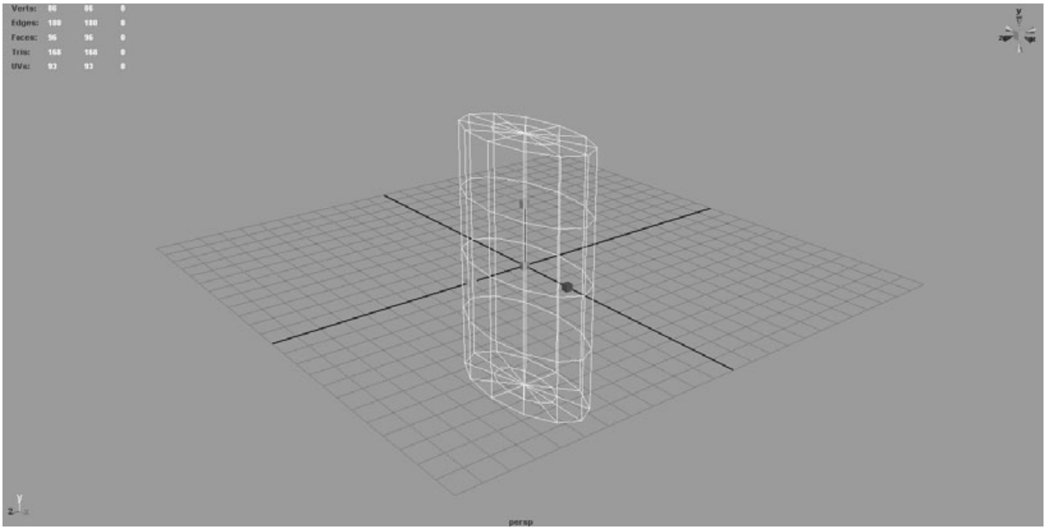


Figure 4.59
Scale the torso down in the z axis.

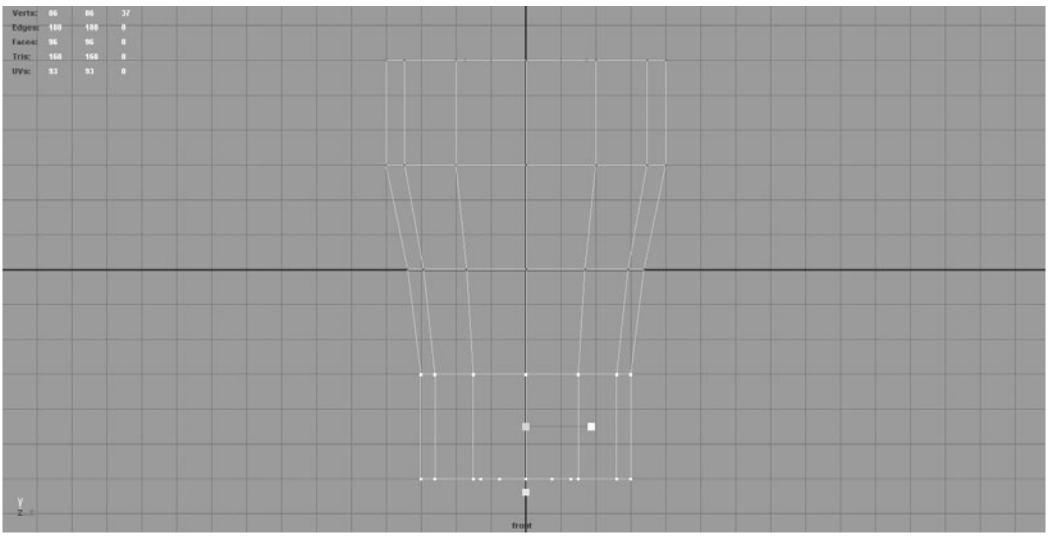


Figure 4.60
Make the character narrower in the hips.

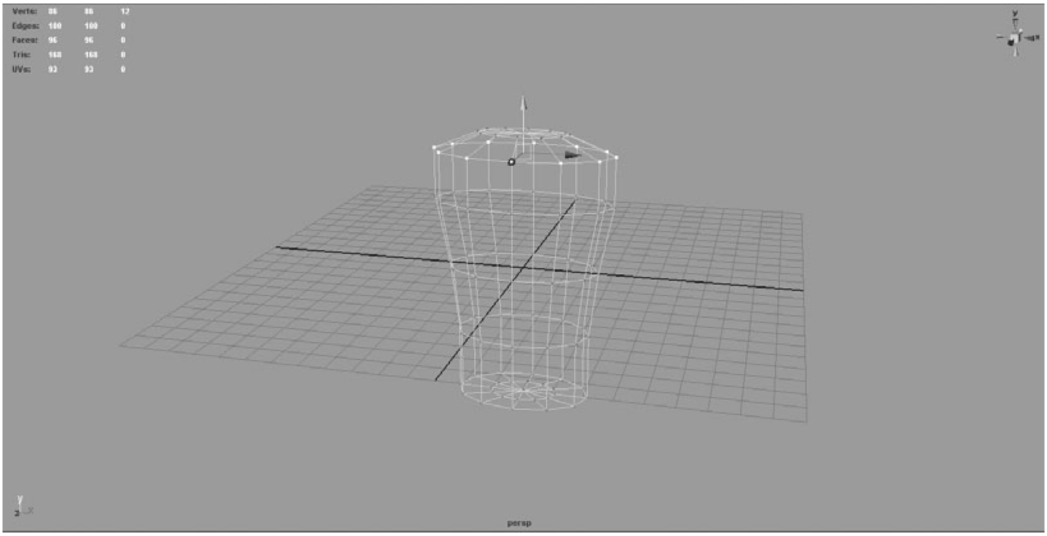


Figure 4.61
Round the character's shoulders.

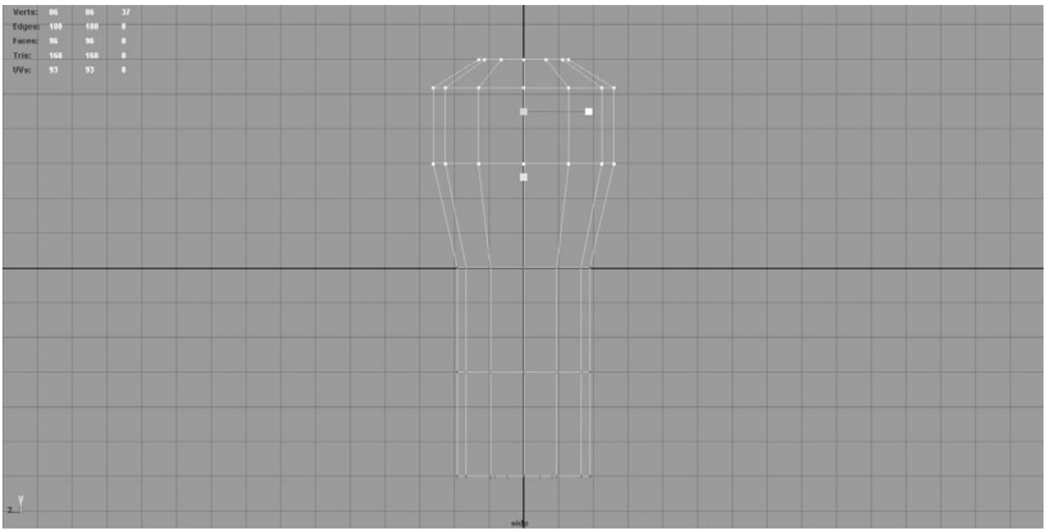


Figure 4.62
Make the character thicker through the chest.

6. Expand the character in the hips as shown in Figure 4.63.

7. Adjust the vertices where the character's arms attach to the torso as shown in Figure 4.64. (Select the vertices by holding the left mouse button down and dragging around the vertices so that you select both the vertex on the plane facing you and the vertex on the surface facing away.)

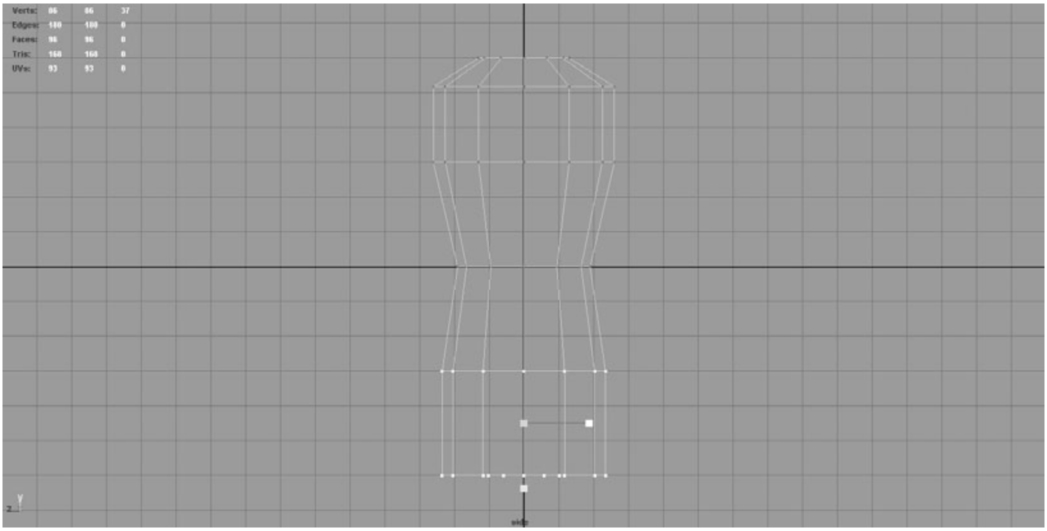


Figure 4.63
Make the character thicker through the hips as well.

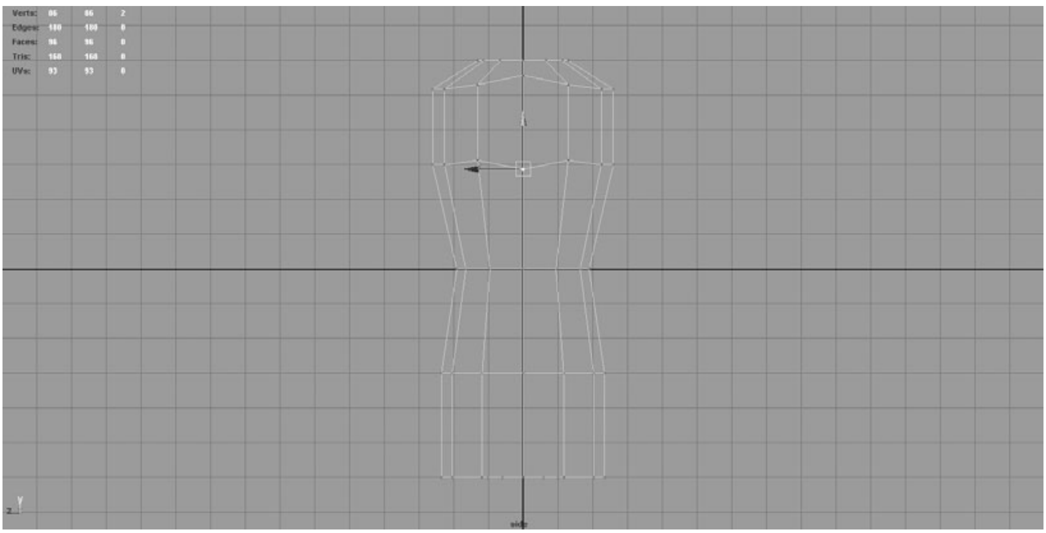


Figure 4.64
Adjust the vertices of the shoulder.

8. Select the front vertices of the chest and pull them outward, as shown in Figure 4.65, to give the character's chest and torso a more natural shape from the side.

9. In Perspective view, select the vertices at the base of the character's neck. Scale the vertices in to better fit the shape of the neck as shown in Figure 4.66.

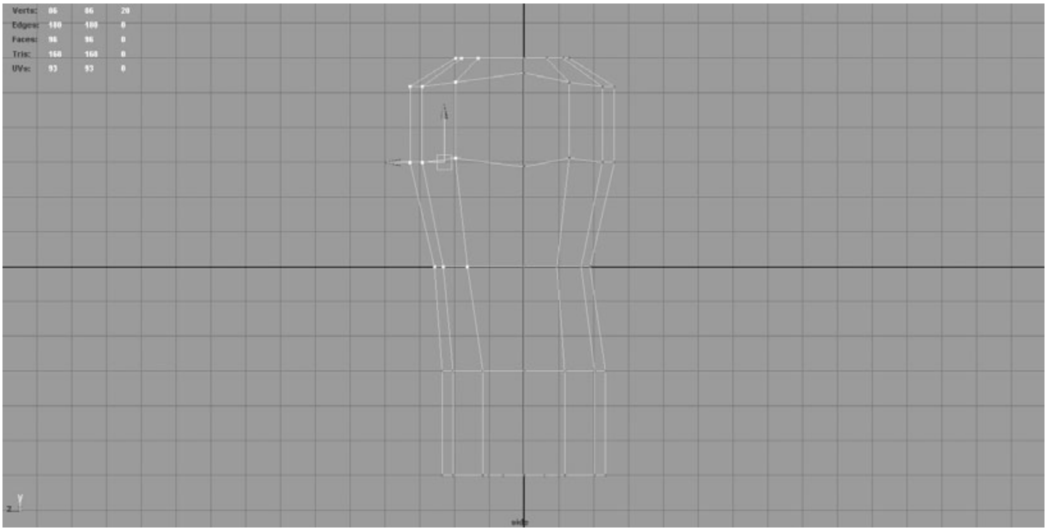


Figure 4.65
Pull the chest forward.

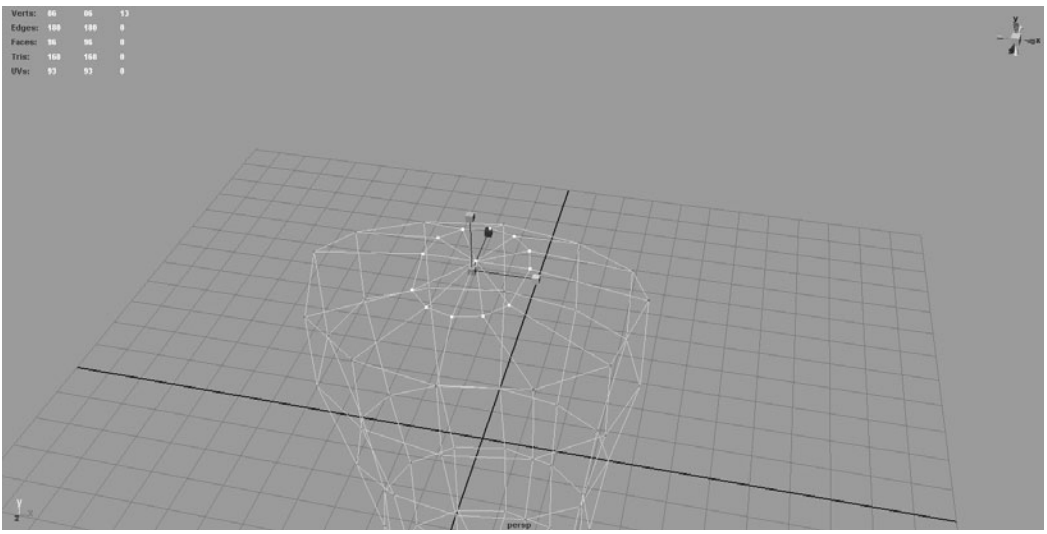


Figure 4.66
Scale the neck area of the character.

10. To begin to form the area where the legs attach to the body, select the vertices of the inner ring at the bottom of the torso and scale them together in the x axis as shown in Figure 4.67.

11. Use the Scale tool to adjust the vertices as shown in Figure 4.68 for the inside of the character's leg.

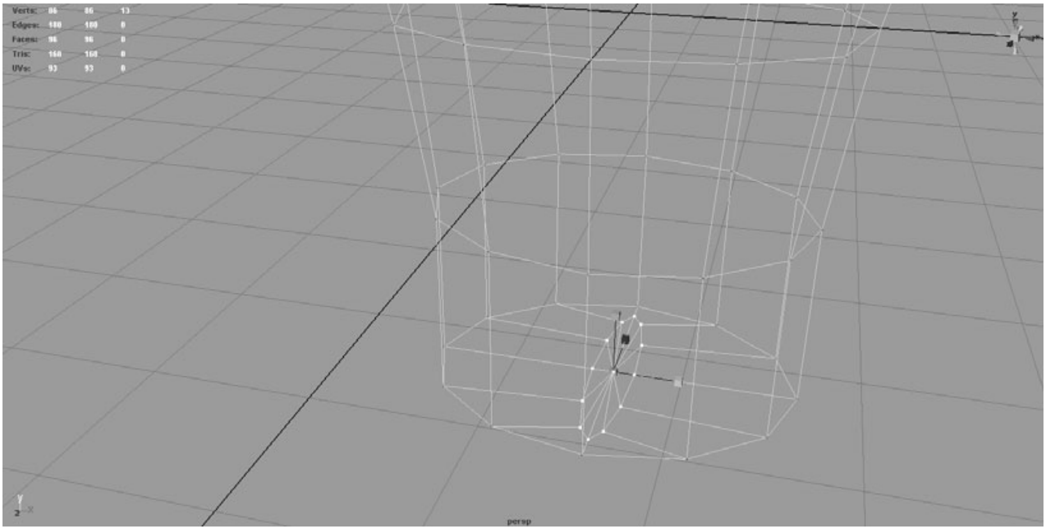


Figure 4.67
Scale the vertices at the bottom of the torso.

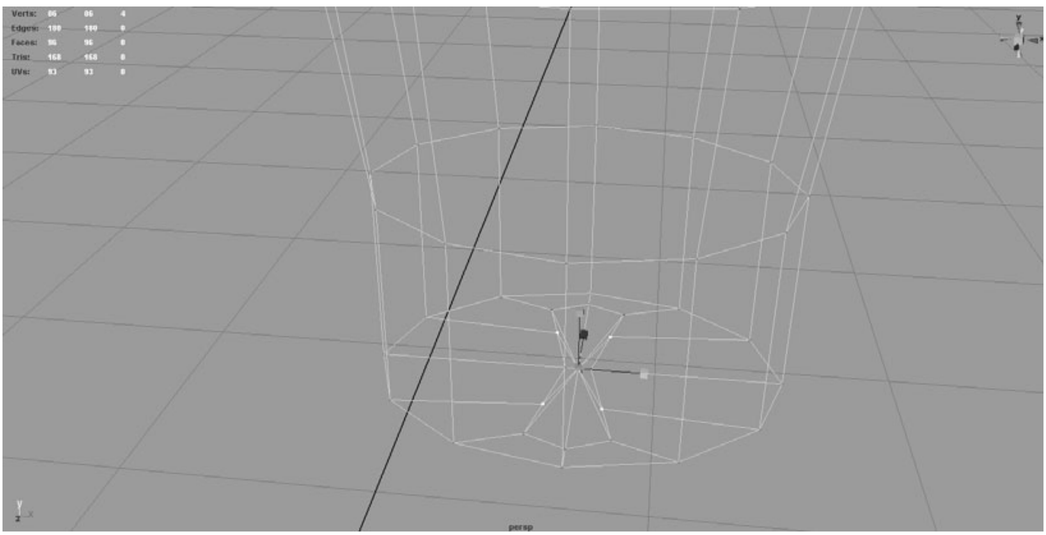


Figure 4.68
Adjust the vertices along the inside of the leg.

12. Select the newly adjusted vertices and use the Move tool to pull them down as shown in Figure 4.69.
13. The hip area is now too long. Use the Move tool as shown in Figure 4.70 to move them up and give the torso a better shape.

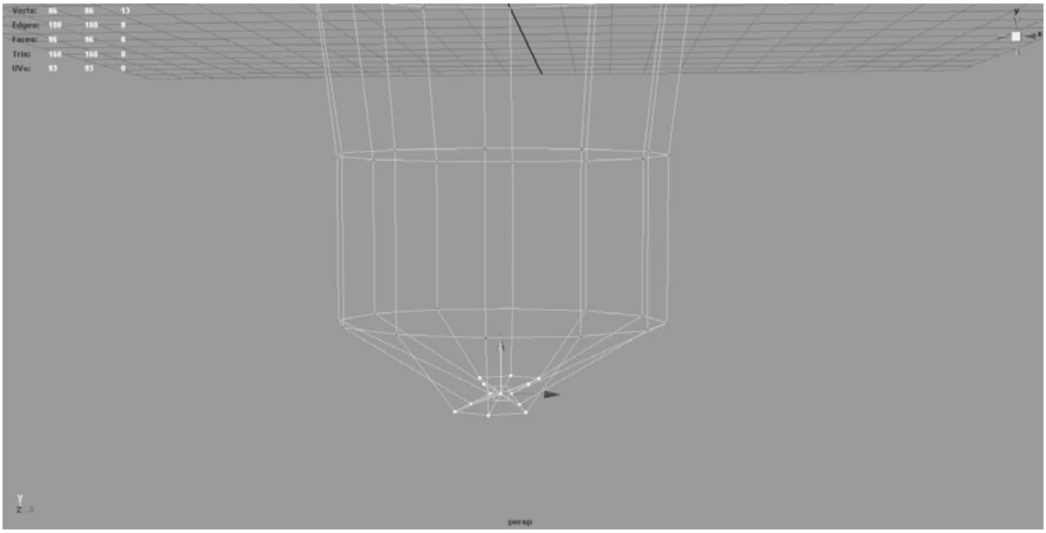


Figure 4.69
Pull the character's groin area downward.

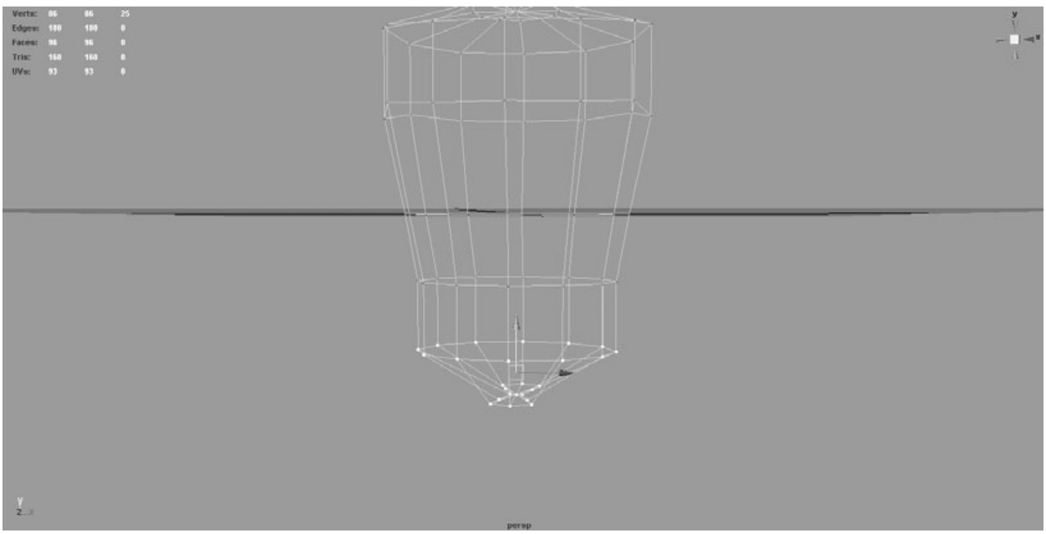


Figure 4.70
Pull the character's hips upward.

The torso is not quite finished, but it is close enough to move on to the character's head. The method you used to create this basic torso is not too much different from the approach you'd take to create a more complex character. The main differences would be the number of polygons and the detail that you would put into the character. A more complex character, for example, would have more detail in the shape of the chest and include definition for the major muscle groups.

Building the Head

Although you *could* extrude the head from the torso—many artists create their character's heads that way—for this project you'll use another primitive shape: a sphere. You'll find that this approach is a bit faster than the extrusion method.

1. Create a polygon sphere with a radius of 2 units and a height of 4 units. Set the subdivisions around the axis to 12 and the subdivisions along the height to 8, as shown in Figure 4.71.

2. Position the sphere above the torso as shown in Figure 4.72.

3. Change the selection mode to Vertex and select the three center rows of vertices. Then scale the selected vertices together in the x axis as shown in Figure 4.73.

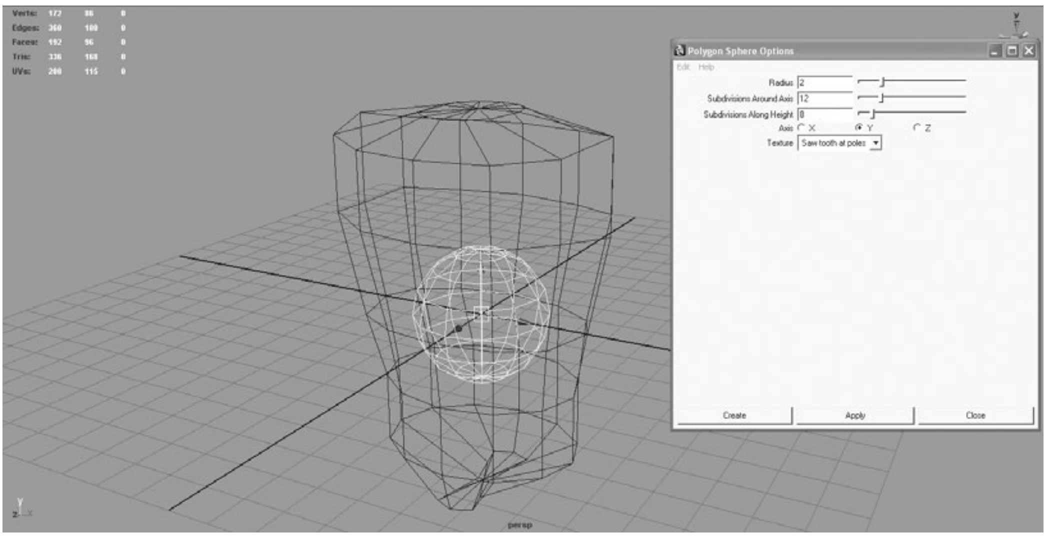


Figure 4.71
Create a polygon sphere for the head.

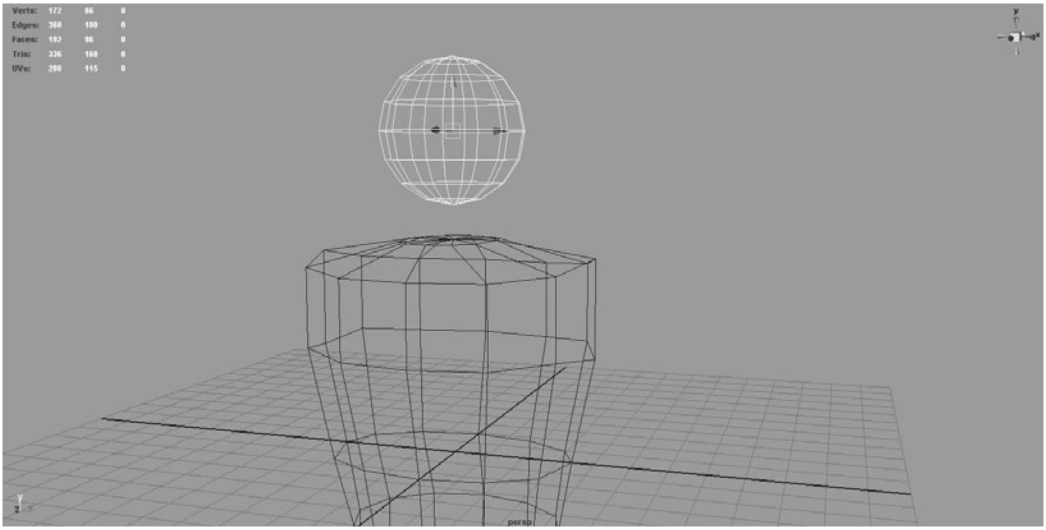


Figure 4.72
Move the sphere above the torso.

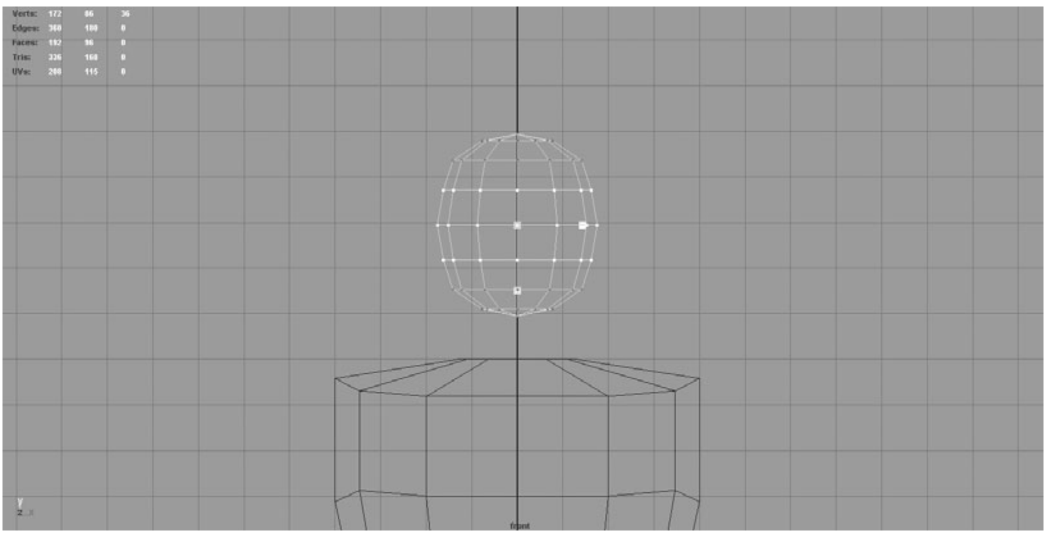


Figure 4.73
Make the head narrower.

4. In Side view, select the vertices of the front of the head and scale them together to flatten the front of the face as shown in Figure 4.74.

5. Change the selection mode to Face and select the faces of the flattened area. Extrude the faces and pull them forward as shown in Figure 4.75.

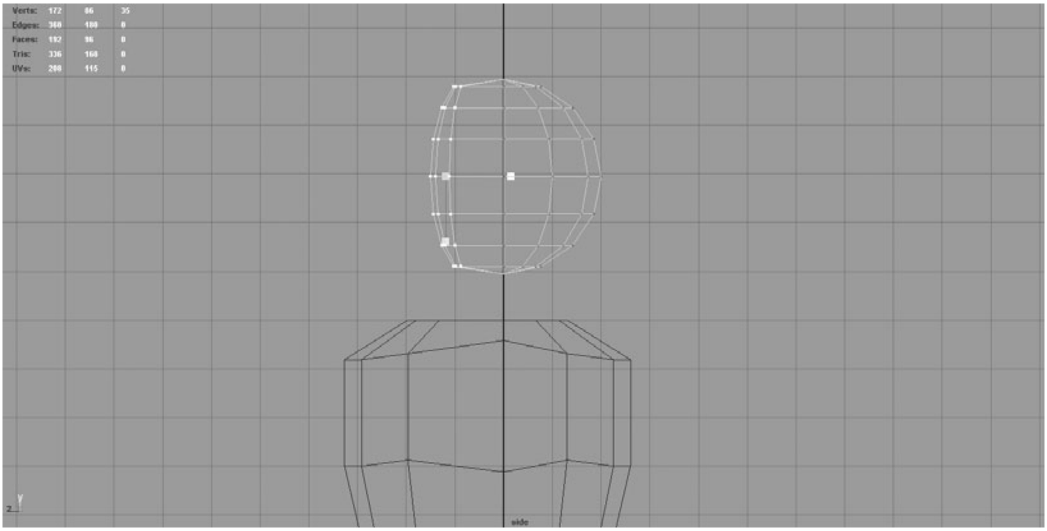


Figure 4.74
Flatten the front of the face.

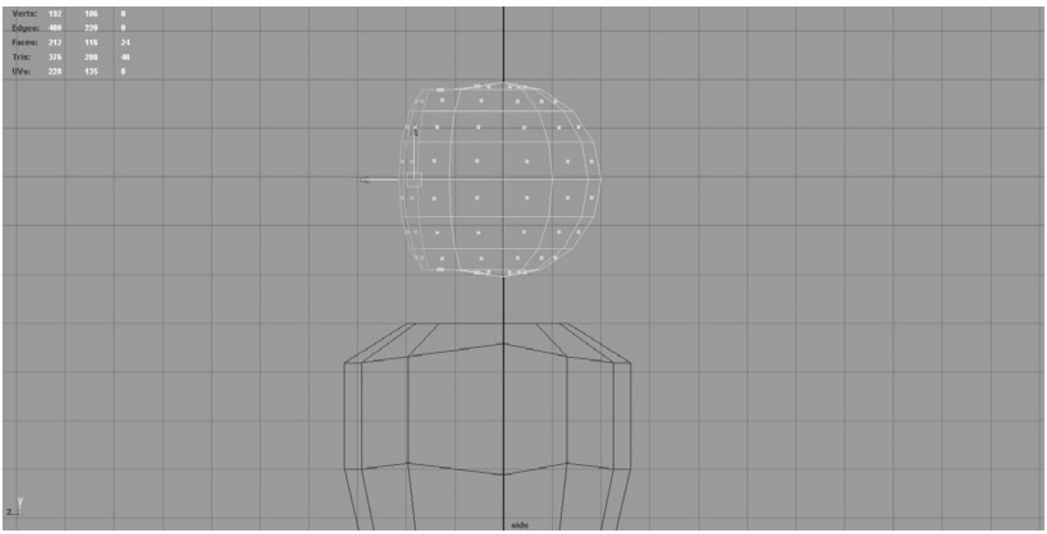


Figure 4.75
Extrude the front faces of the head.

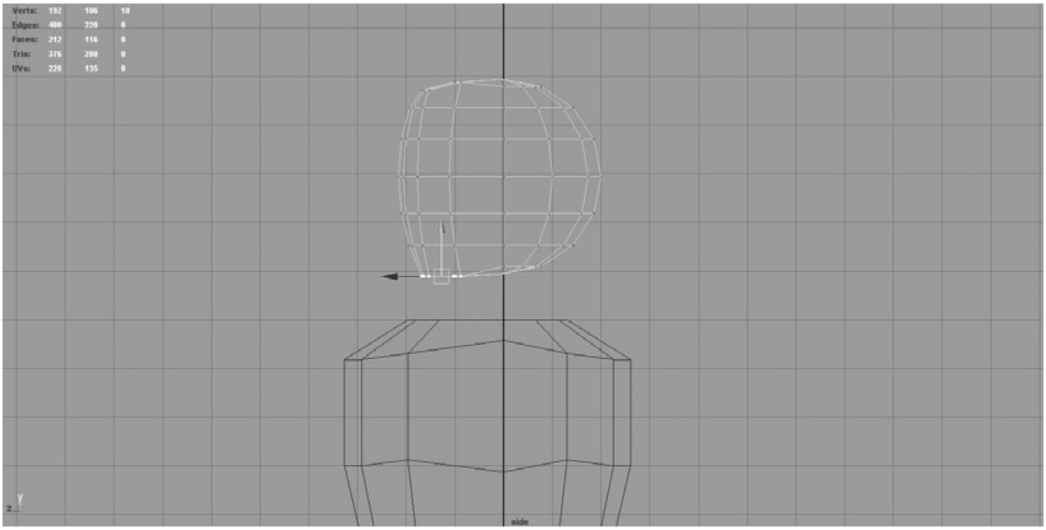


Figure 4.76
Move the vertices to form the character's chin.

6. Change the selection mode to Vertices and move the vertices on the lower front of the face to form the character's chin as shown in Figure 4.76.
7. The front of the sphere needs to be rounded more to look more like an actual human head. Use the Scale tool to make adjustments to the vertices. (The Scale tool works well when you want to keep you adjustments symmetrical.) See Figure 4.77 for an example of the results of the scaling adjustments.
8. Now that the basic shape of the head is finished, it's time to work in the features. To begin, change the selection mode to Faces and select the faces of the polygons that will form the forehead and nose of the character. Extrude these faces forward as shown in Figure 4.78.
9. Use the Scale tool to adjust the vertices of the nose as shown in Figure 4.79.
10. Pull out the vertices of the bridge and tip of the nose and adjust the vertices of the brow as shown in Figure 4.80.

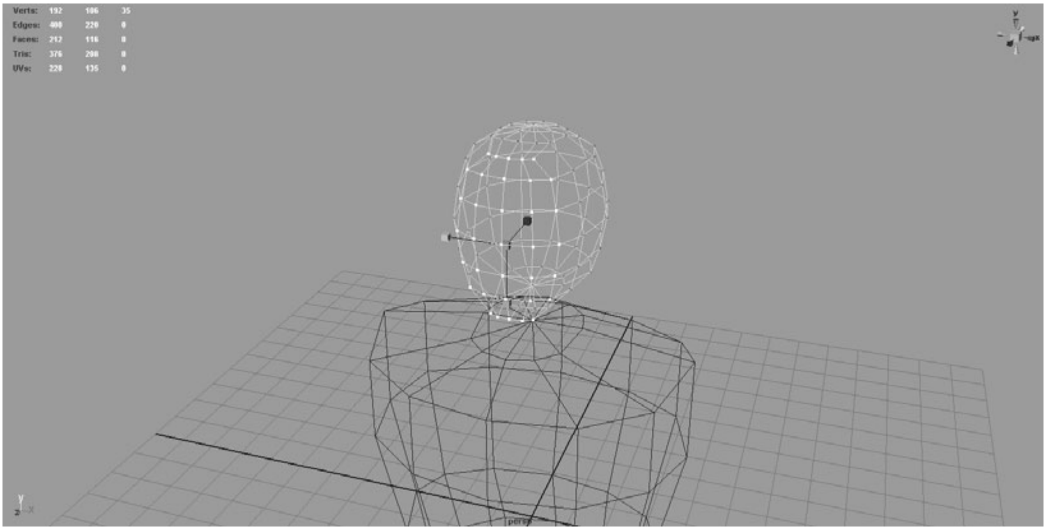


Figure 4.77
Round some of the corners of the head with the Scale tool.

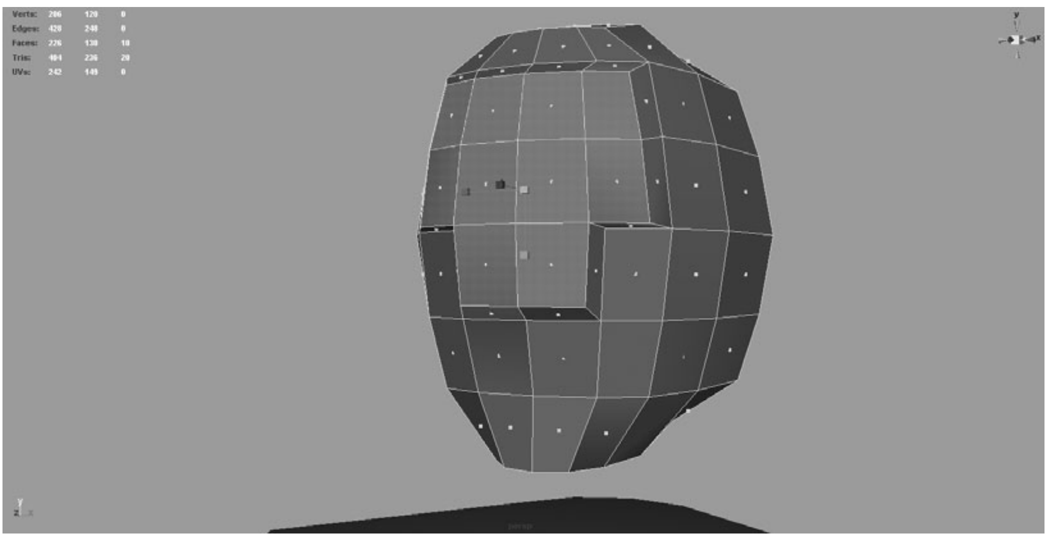


Figure 4.78
Extrude the faces of the forehead and nose.

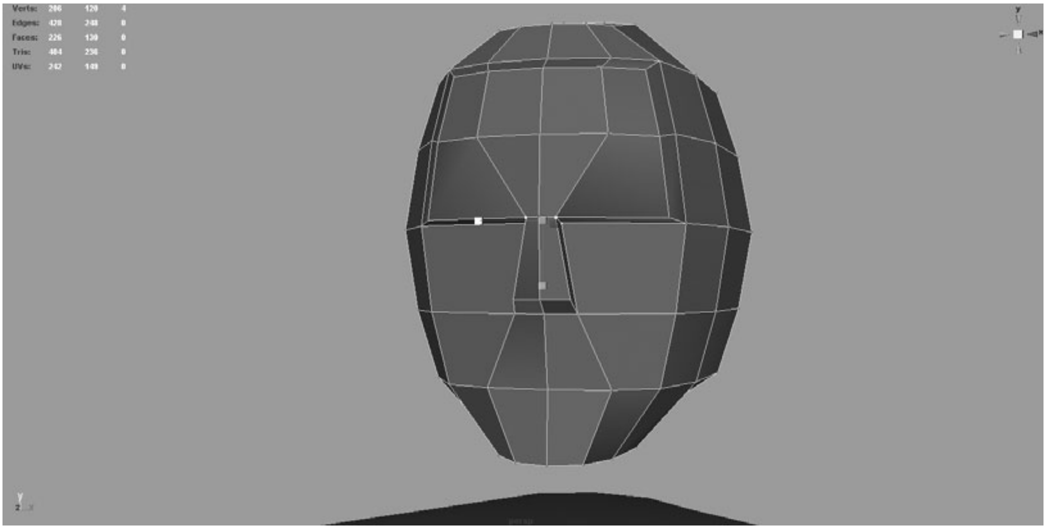


Figure 4.79
Scale the vertices to shape the nose.

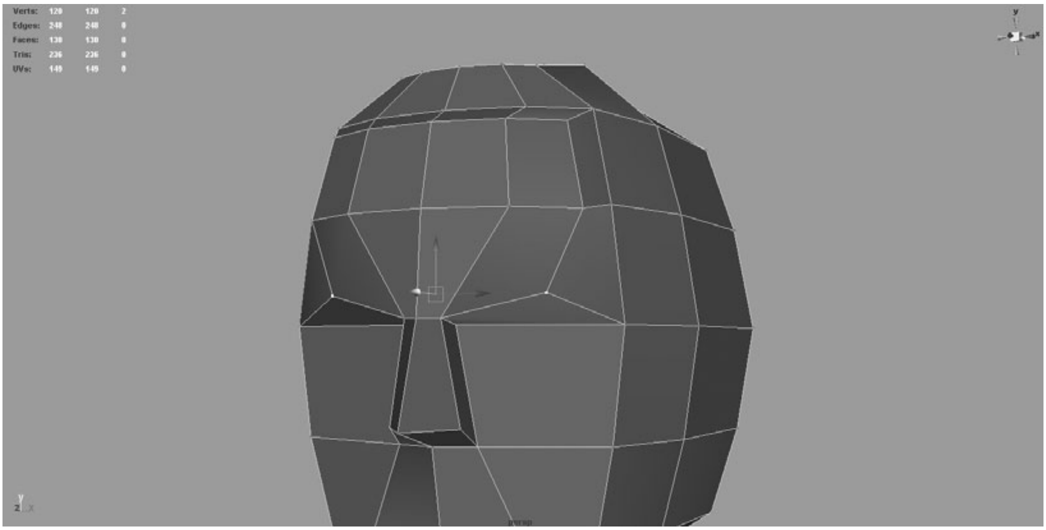


Figure 4.80
Shape the character's brow.

11. Move the vertices of the top of the head forward to form a hairline (see Figure 4.81).

You now have the basic shape for your character's head. There are still elements of the head that are yet incomplete, like the mouth and ears, but the basics are in place. For now, move on to the rest of the character.

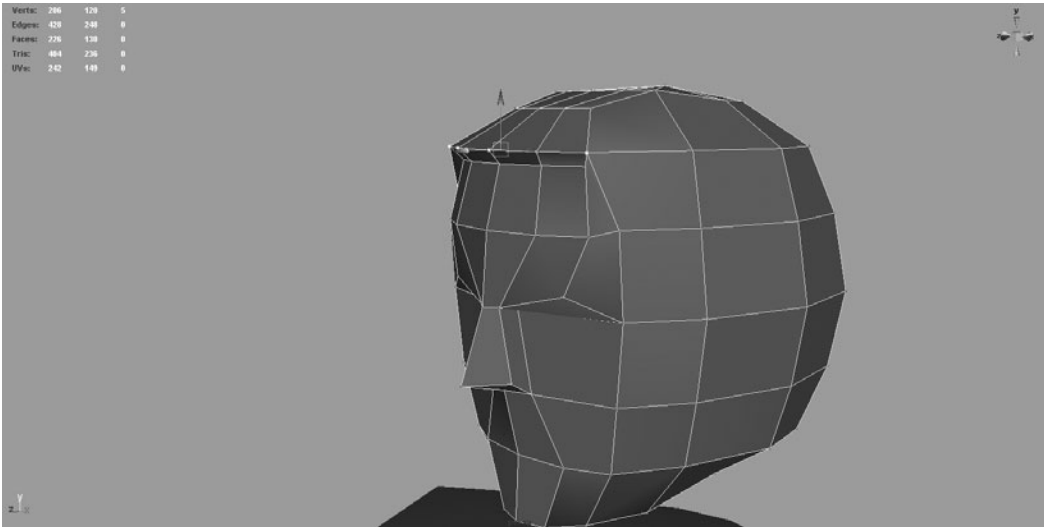


Figure 4.81
Give the character a hairline.

Note

Game characters often rely on textures rather than geometry for some of the detail of the face and head. Because this is just a simple model to get you started, you'll skip that process here. You'll work on a more complex model in the next chapter.

Attaching the Head

In games, it is often desirable for the character's geometry to be one continuous mesh because single-mesh geometry tends to deform better when animated. Right now, however, the character's head and torso are two separate objects. In this section, you will see how to combine the two separate objects into a single continuous mesh.

1. Select the bottom ring of faces on the head and extrude the character's neck down as shown in Figure 4.82.
2. Delete the faces at the bottom of the neck and the corresponding faces in the top of the torso to create a hole in both the torso and neck.
3. Change the selection mode to Object and select both objects.
4. From the Mesh menu, select Combine (see Figure 4.83) to combine the selected objects into a single object.

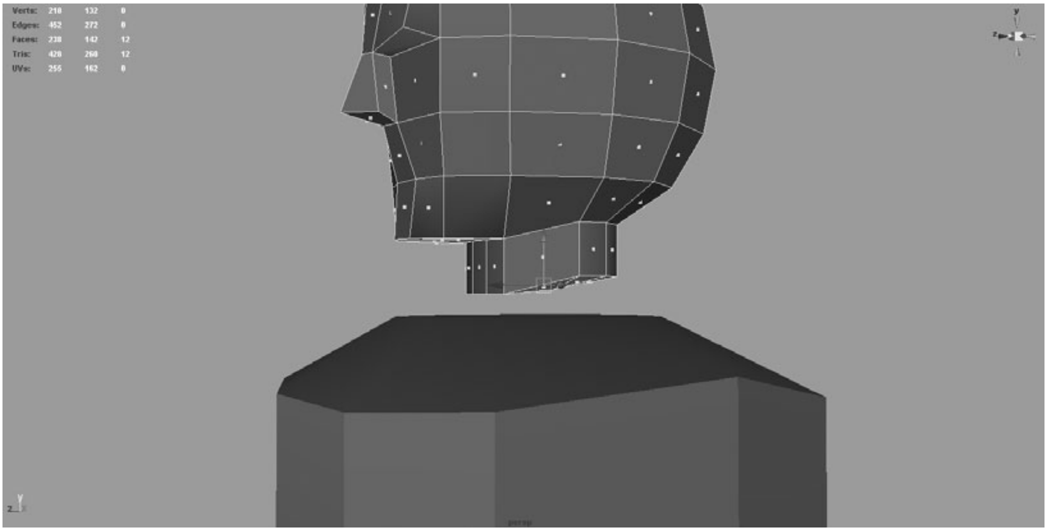


Figure 4.82
Extrude the neck from the head.

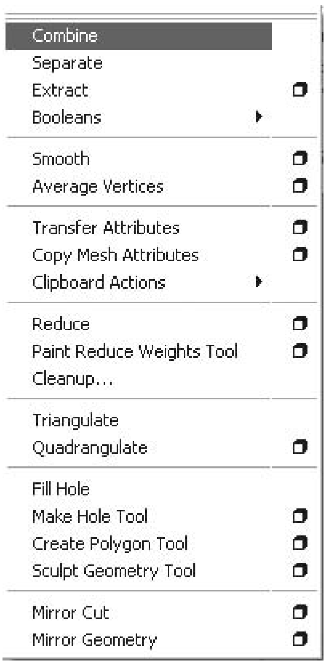


Figure 4.83
Combine the two objects into a single object.

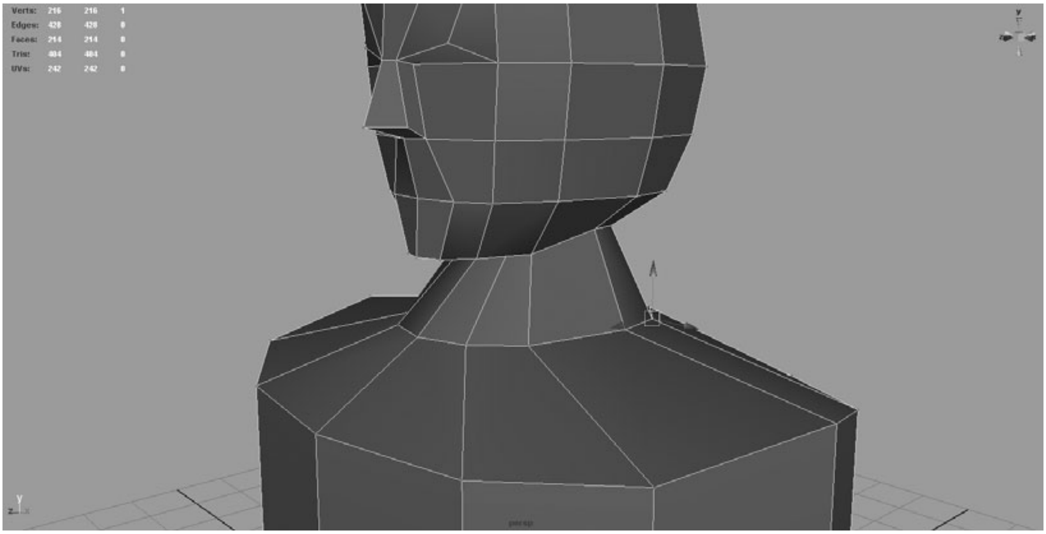


Figure 4.84
Snap the vertices of the neck to the torso.

5. Switch back to Vertex selection mode and press Ctrl+V while using the Move tool to snap the vertices of the neck to the corresponding vertices of the torso, as shown in Figure 4.84. (Because the subdivisions around the head and torso were originally set to 12, the two should match.)
6. When the vertices are snapped together, merge them by opening the Edit Polygons menu and selecting Merge Vertices as shown in Figure 4.85.
7. In Side view, move the head forward to better follow how the head should attach to the body as shown in Figure 4.86.
8. The head is just a little small for the torso. To remedy this, select the vertices of the head and scale them up a little to better match the torso, as shown in Figure 4.87.

Now the head and the body are connected into a single mesh. Your character now needs arms and legs.

Building the Arms and Legs

Your character will have two symmetrical arms and two symmetrical legs. Rather than modeling an identical piece of geometry twice, instead model half the character and then mirror the geometry to the other side of the character. To

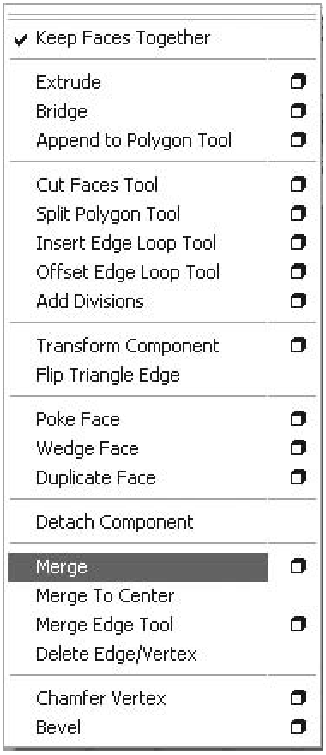


Figure 4.85
Merge the vertices of the neck and torso.

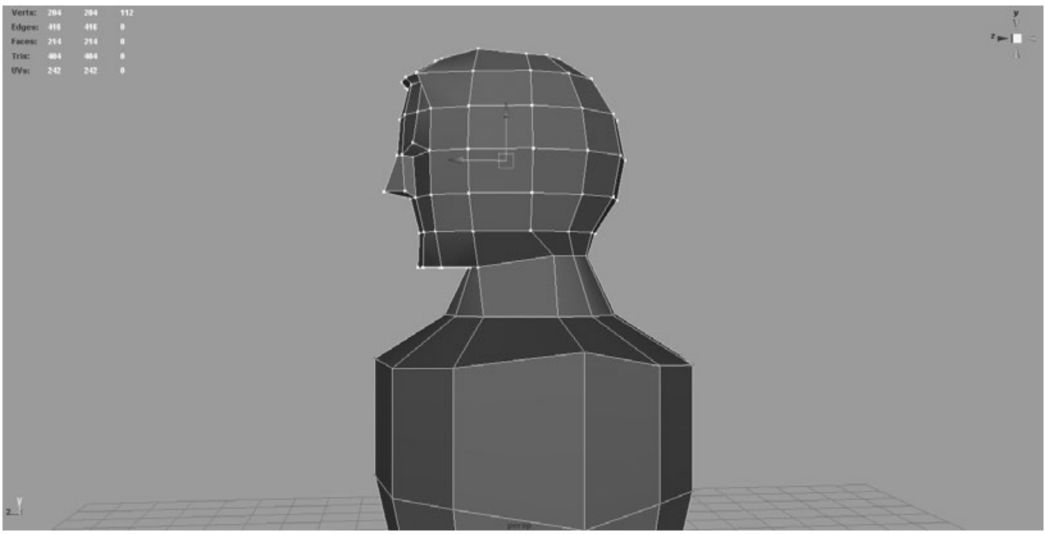


Figure 4.86
Move the head forward.

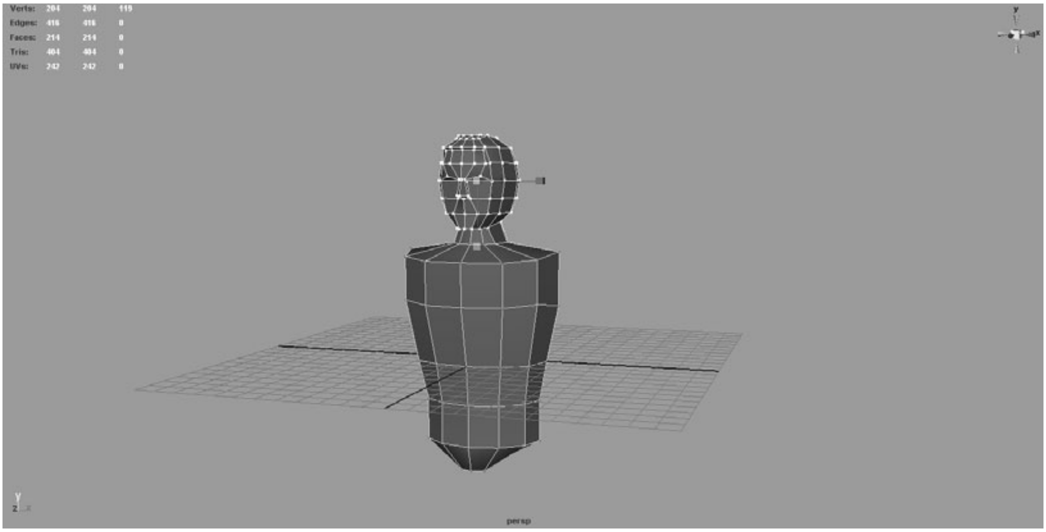


Figure 4.87
Make the head a little larger.

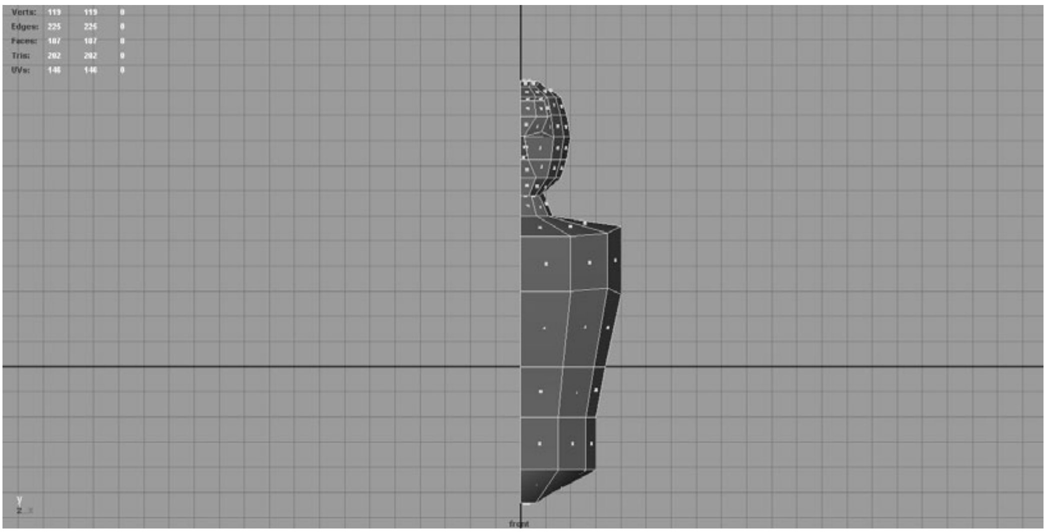


Figure 4.88
Delete the faces of half the model.

prepare the model for the mirror operation, delete half of the current model from the front view as shown in Figure 4.88.

Now the model is ready for the arms and legs to be created. Remember where you adjusted the polygons of the shoulder for the arm to attach? Select those two polygons and extrude them several times to extend an arm from the body as

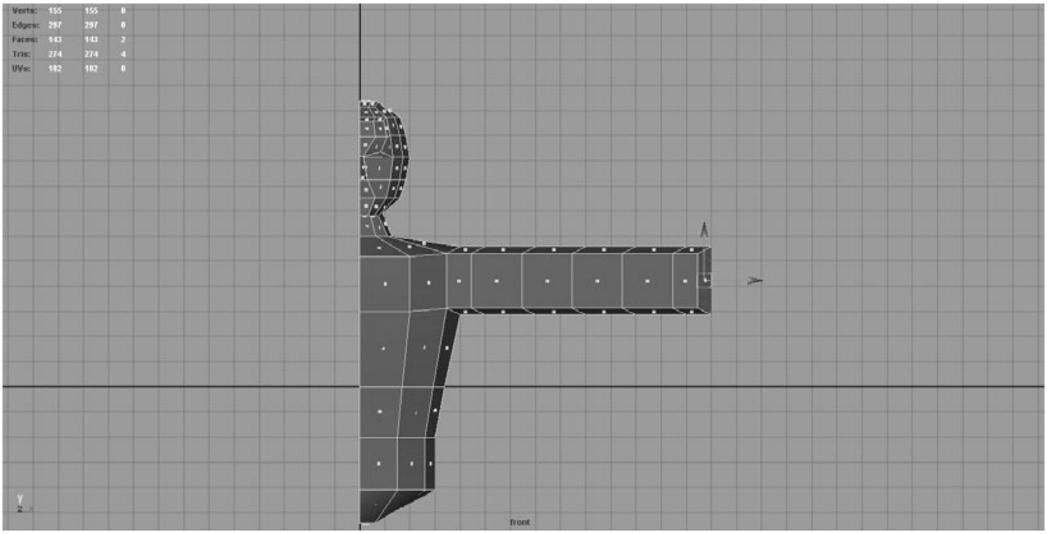


Figure 4.89
Extrude faces to form the character's arm.

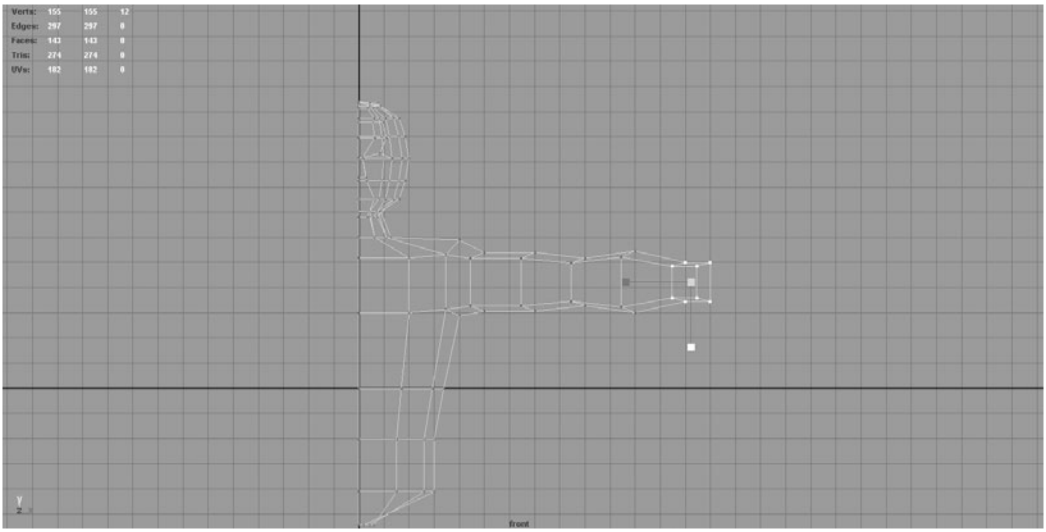


Figure 4.90
Give the arm a better shape from the Front view.

shown in Figure 4.89. These extruded polygons will form the basis for the character's left arm.

Right now the arm looks kind of blocky. To shape the arm so it looks a bit more natural, do the following:

1. In Front view, use the Scale and Move tools to give the arm a more natural look (see Figure 4.90).

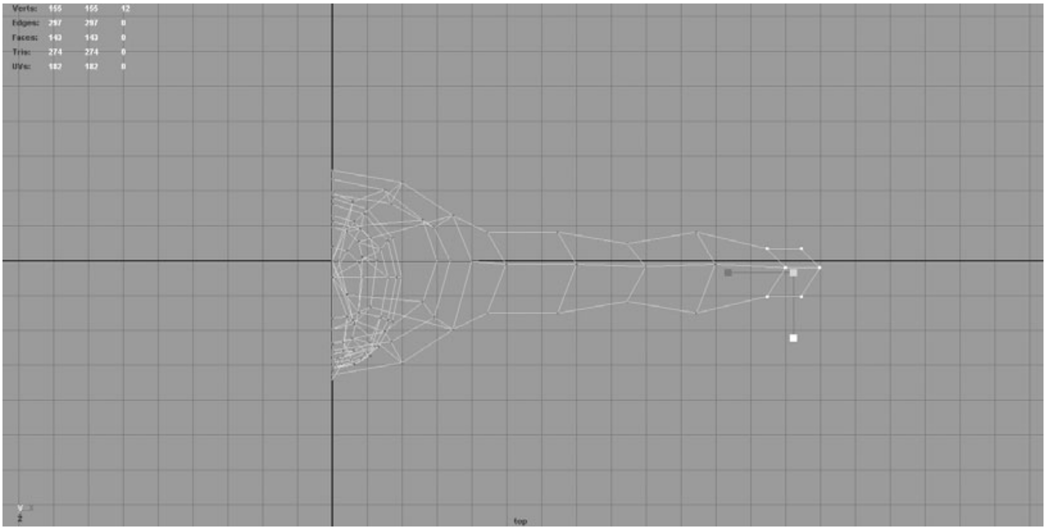


Figure 4.91
Adjust the arm from the top view.

2. Adjust the arm from the top view, as shown in Figure 4.91.
3. Switch to Perspective view so you can freely look at the arm from any angle (see Figure 4.92).

The character has an arm; now he needs a leg. The leg is a little more complex in its attachment to the body than the arm. To animate well, the leg needs to be attached at an angle even though it will be extruded straight down.

1. Move the vertices of the hip area up to form an angle for the leg to attach to the body as shown in Figure 4.93.
2. Select the faces where the leg will be extruded and extrude them down. Use the Scale tool to change the angle of the extruded polygons as shown in Figure 4.94.
3. In the next extrusion, use the Scale tool to make the extruded polygon surface horizontal.
4. Make several extrusions for the leg, using Figure 4.95 as a guide for placing each extruded section. Notice that they are not all the same distance from each other; this is to better follow the eventual form of the leg.
5. Like you did with the arm, use the Scale and Move tools to give the leg a more natural shape from the front as shown in Figure 4.96.

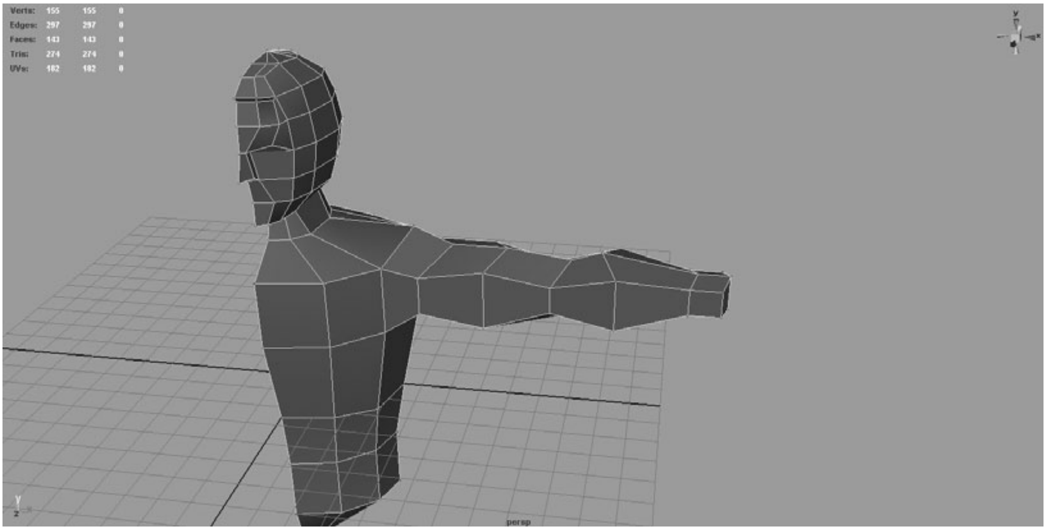


Figure 4.92
Finalize the shape of the arm in the Perspective view.

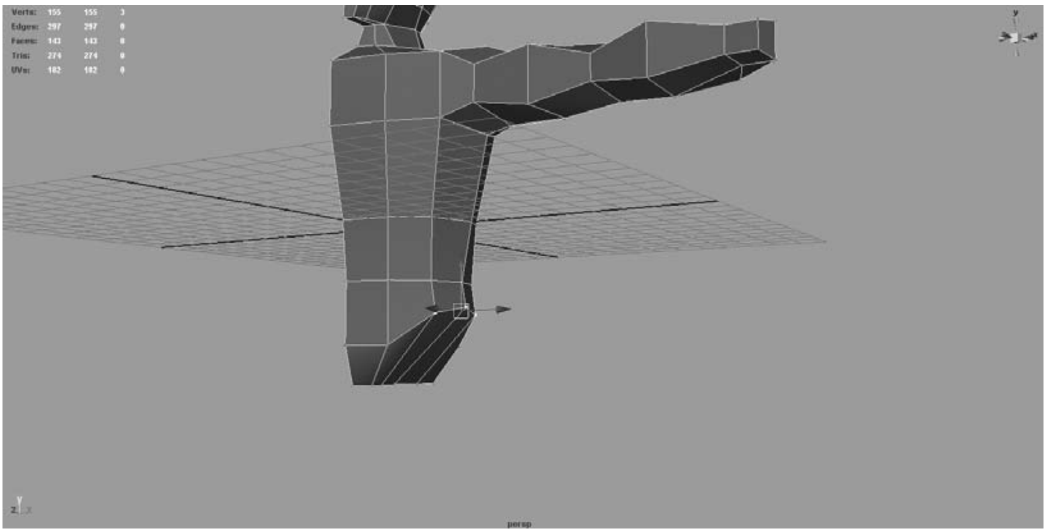


Figure 4.93
Prepare the torso for the leg.

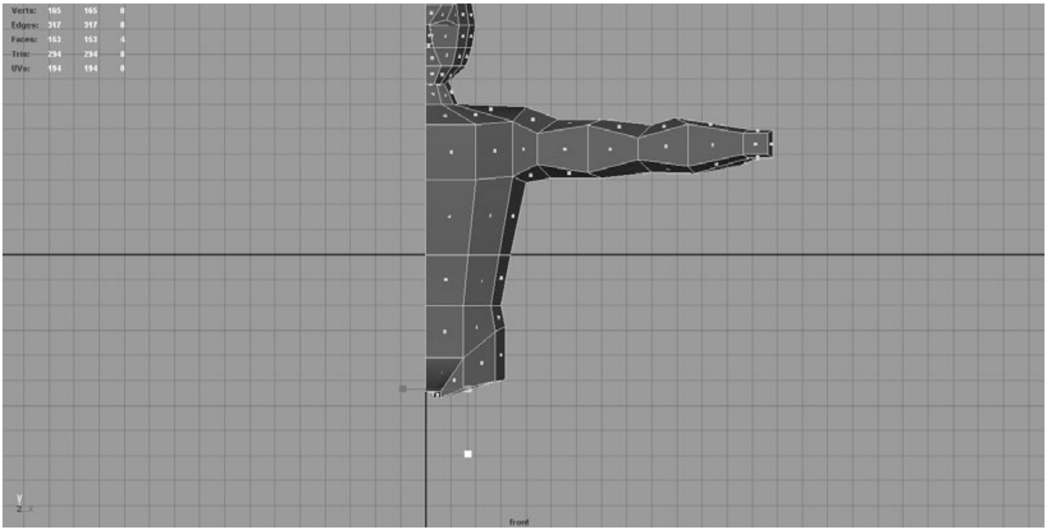


Figure 4.94
Make the first leg extrusion.

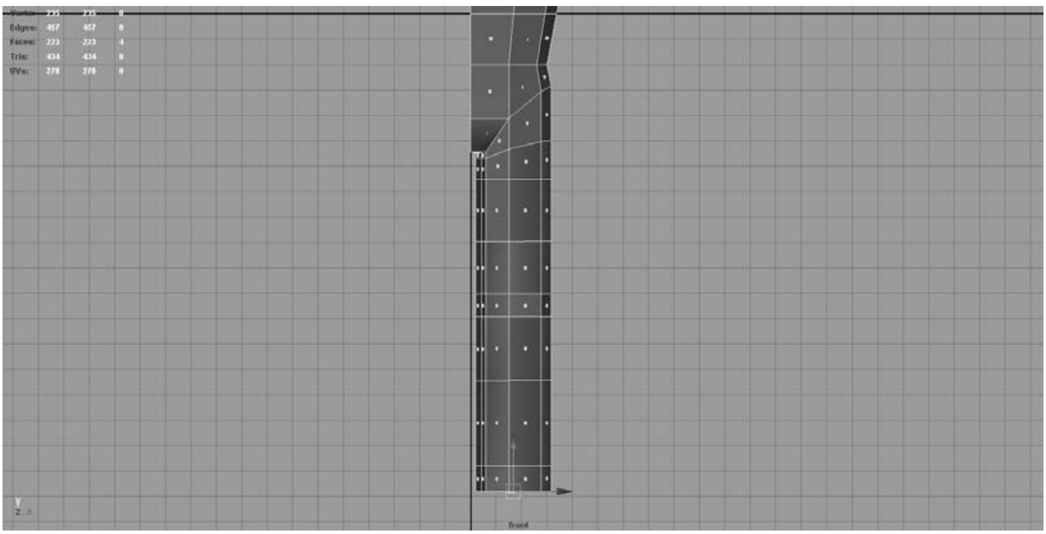


Figure 4.95
Extrude the faces for the leg.

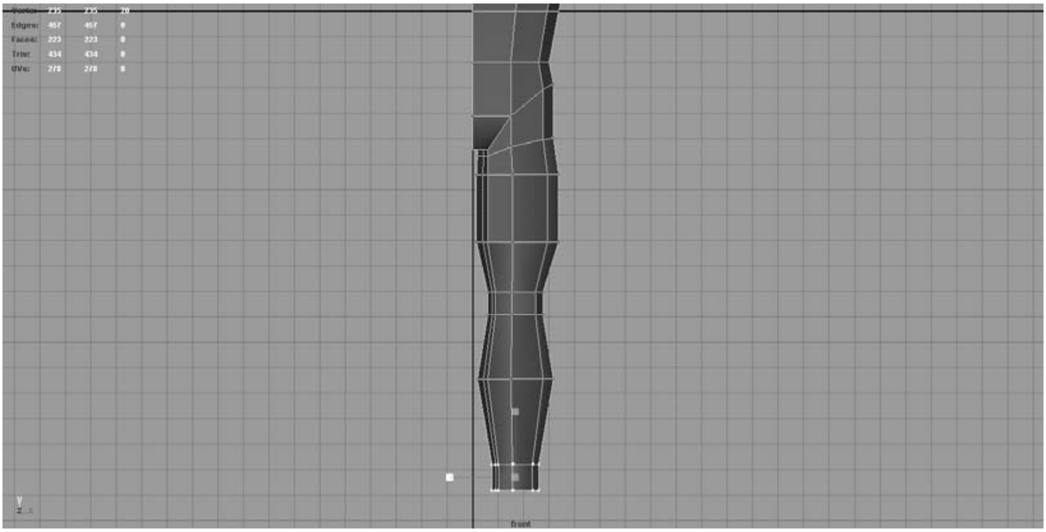


Figure 4.96
Make the leg look more natural from the front.

6. Do the same thing from the side, as shown in Figure 4.97.
7. Finish the shaping of the leg in the Perspective view, as you did with the arm.
8. The leg now needs to be blended with the hips and buttocks area of the body. Adjust the vertices in this region, adding the muscles of the buttocks and hip as shown in Figure 4.98.

The character is nearly finished. It's still very basic, but you can see that it is shaping up nicely. All that is left is to give the character hands and feet.

Creating Hands and Feet

Because this character is a simple character, we'll simplify creating the hands and feet by ignoring the fingers. (Often game characters do not have completely defined fingers and toes to save on polygon counts. This is particularly true for minor game characters.)

1. Select the faces at the end of the arm and extrude them several times as shown in Figure 4.99. Scale the extrusions as shown.
2. In Top view, move the vertices around to follow the shape of the palm and fingers as shown in Figure 4.100.
3. Extrude the thumb from the palm as shown in Figure 4.101.

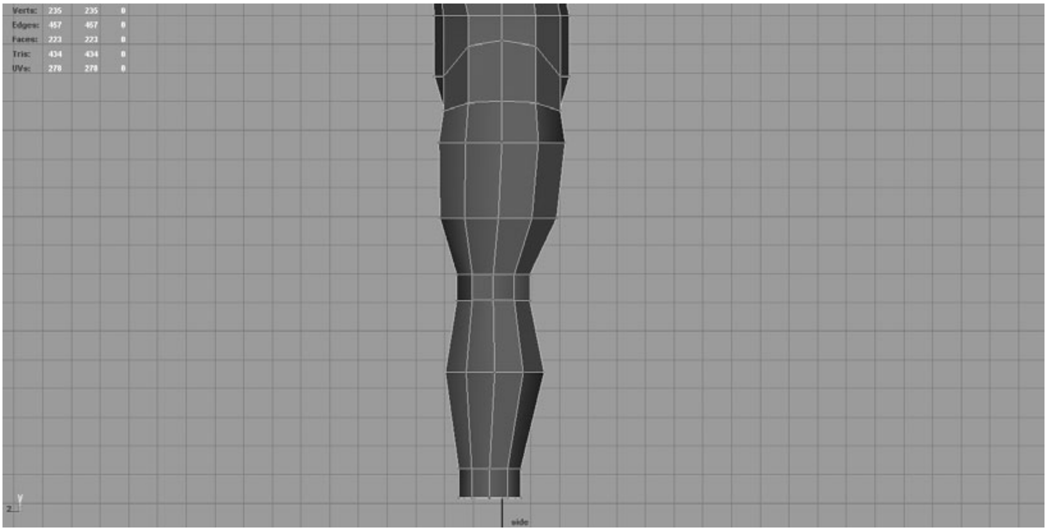


Figure 4.97
Also shape the leg from the side.

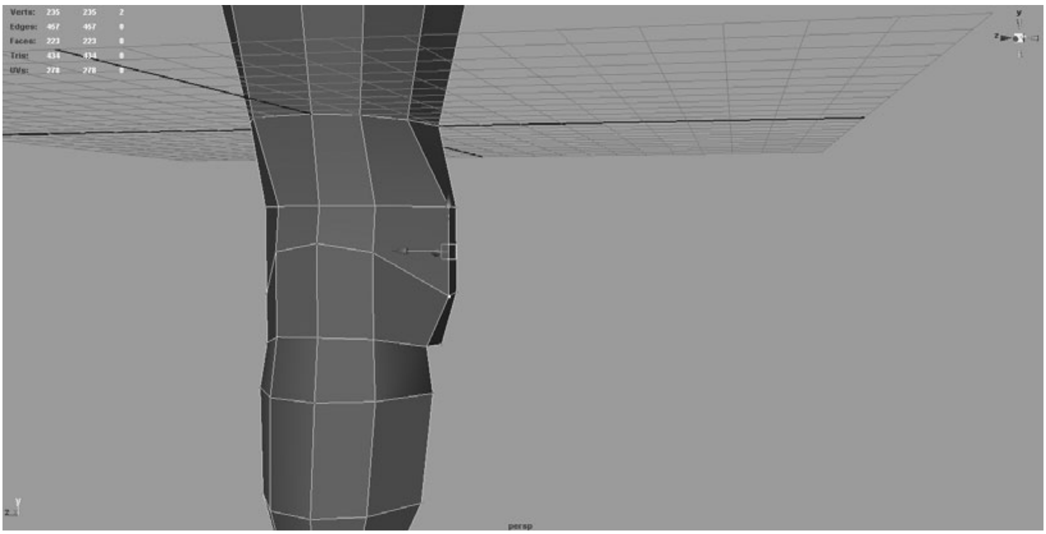


Figure 4.98
Work on the buttocks and hip area.

4. In Perspective view, adjust the thumb so that it is lower than the palm and fingers (see Figure 4.102).
5. In Side view, move down to the end of the leg to work on the foot. Select the faces at the end of the foot and extrude them down; then use the Rotate tool to rotate the extruded faces as shown in Figure 4.103.

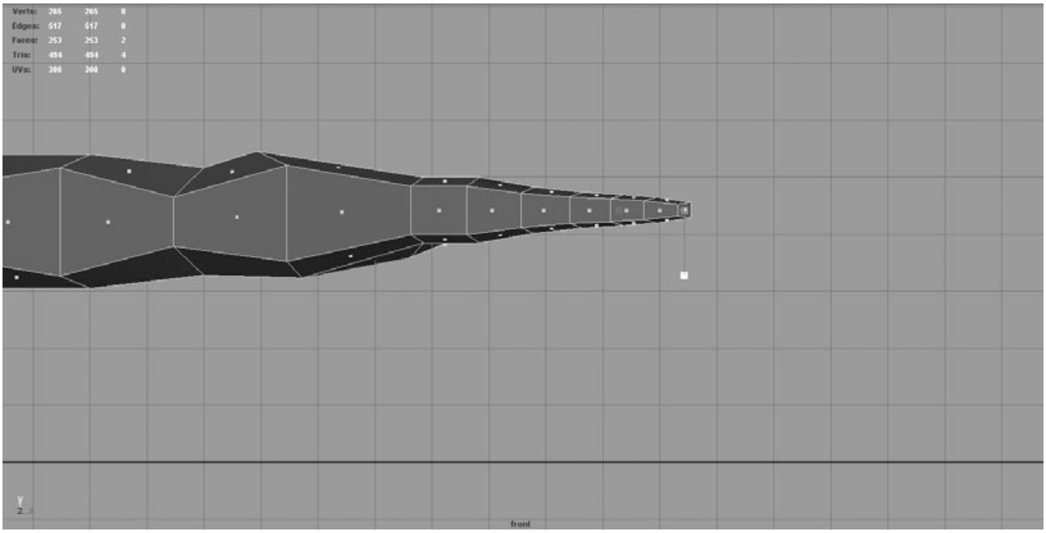


Figure 4.99
Extrude the hand from the arm.

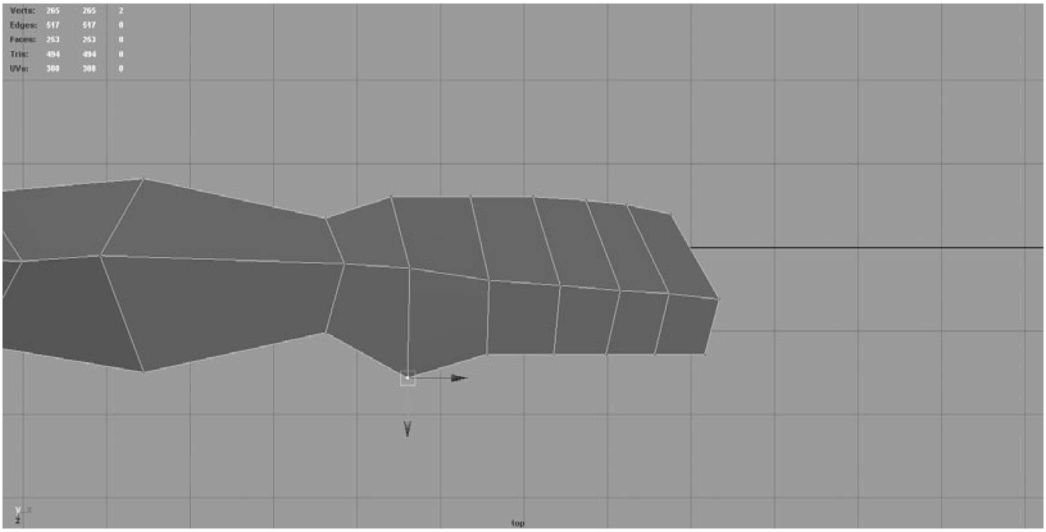


Figure 4.100
Shape the hand from the top view.

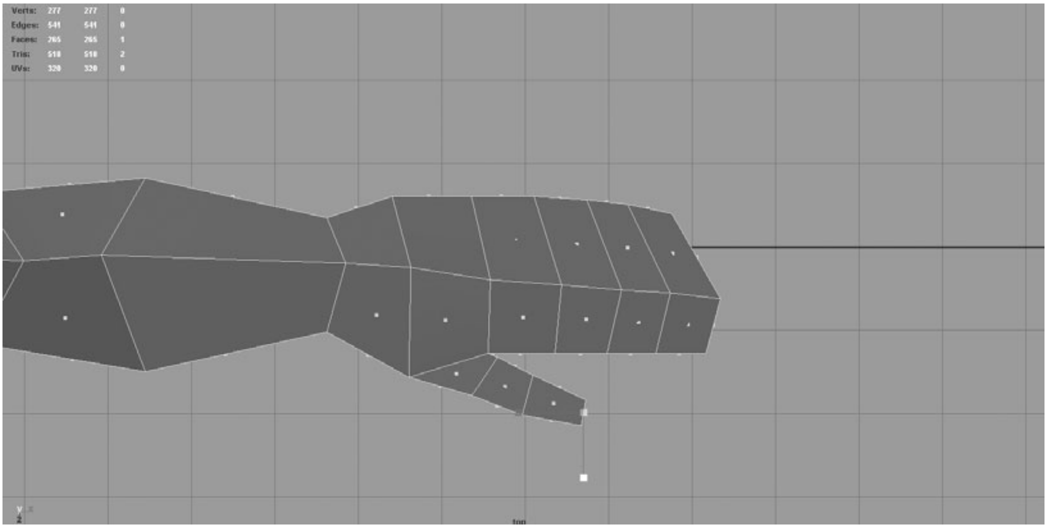


Figure 4.101
Extrude a thumb for the hand.

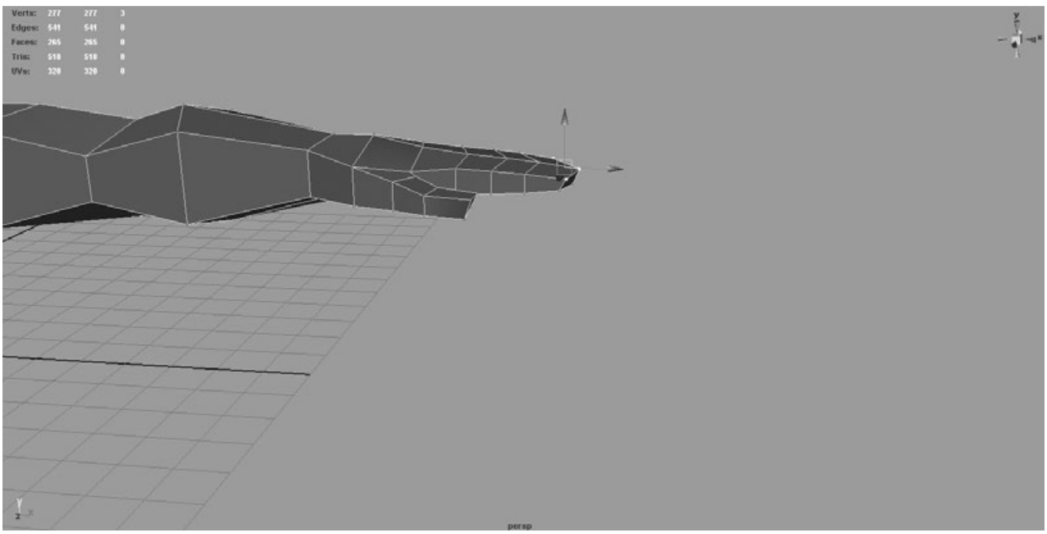


Figure 4.102
Lower the thumb for a more natural-looking hand.

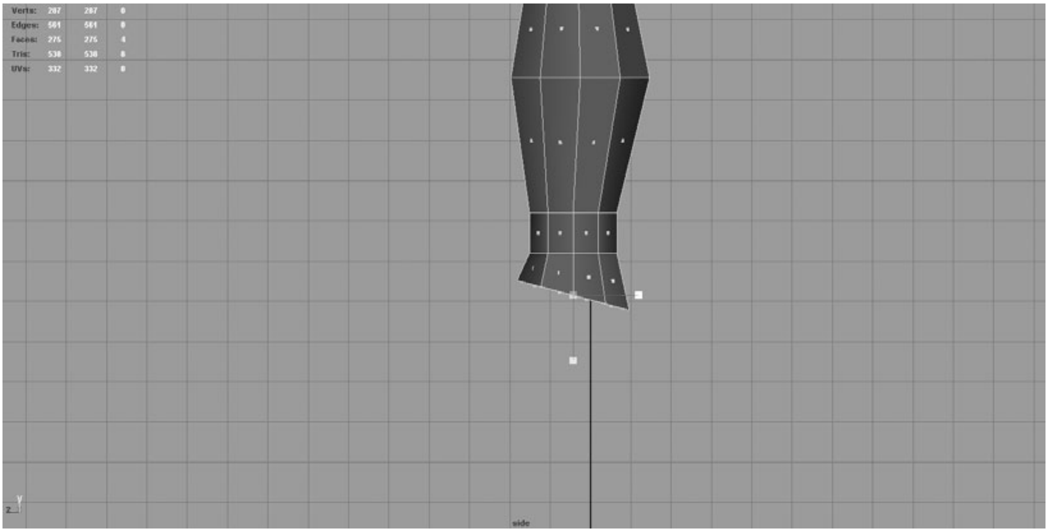


Figure 4.103
Rotate the faces of the foot.

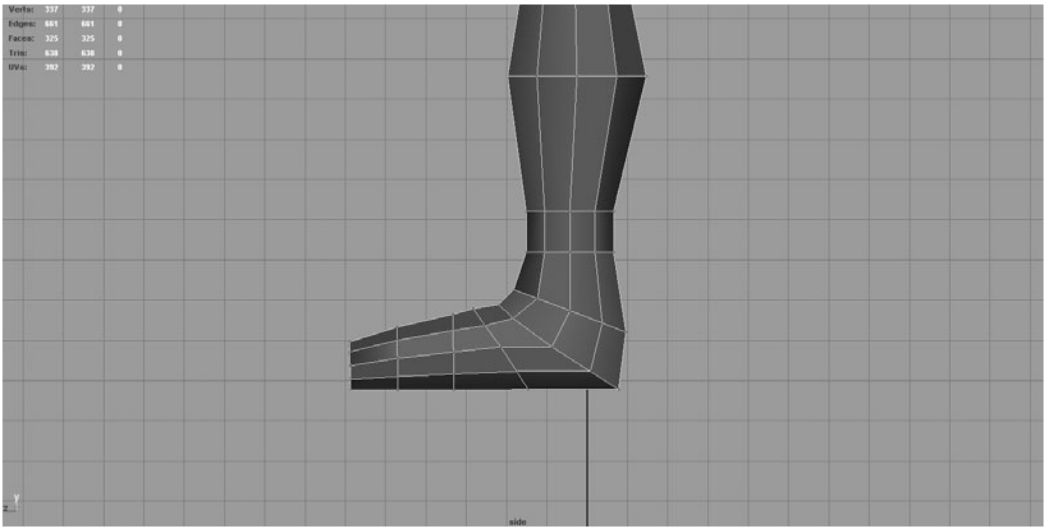


Figure 4.104
Extrude the foot from the bottom of the leg.

6. Continue to extrude the faces of the foot and rotate and scale them as needed (see Figure 4.104).
7. In Perspective view, shape the foot to look more natural (see Figure 4.105).

The model is almost finished; it just needs to be mirrored and smoothed. To do so, return to Object selection mode. Then open the Mesh menu and choose select

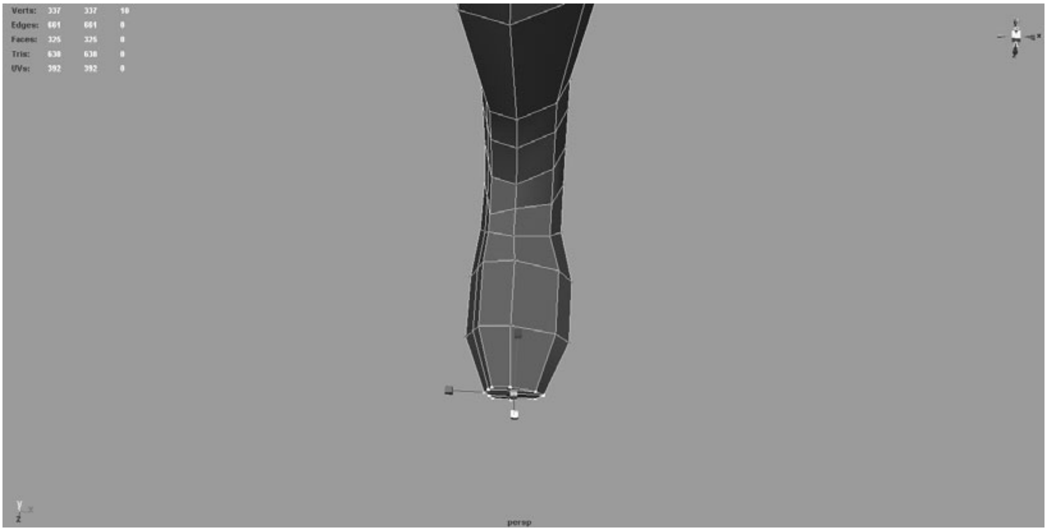


Figure 4.105
Shape the foot from the Perspective view.

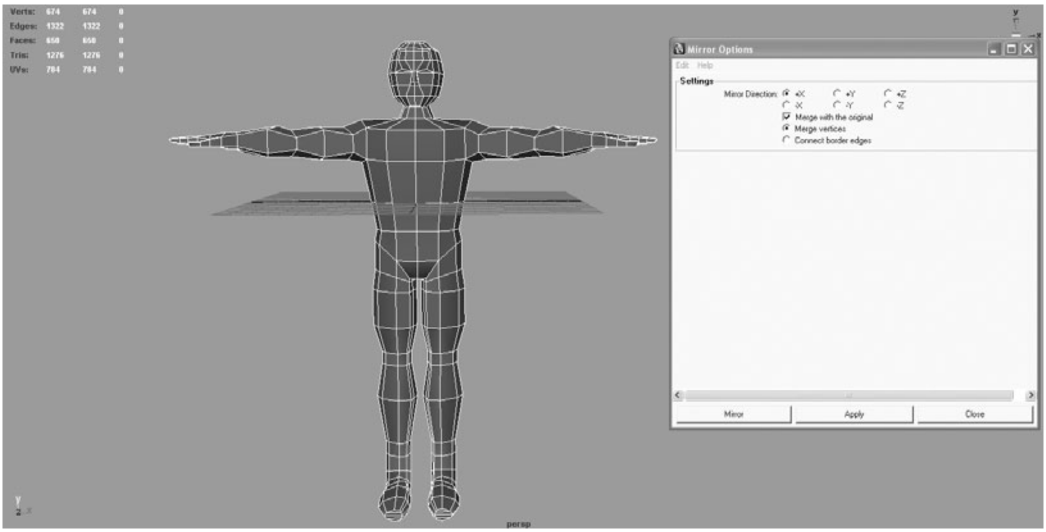


Figure 4.106
Mirror the model across the x axis.

Mirror Geometry. Finally, click the −X option in the Mirror Options dialog box, shown in Figure 4.106

Because of occasional problems, I typically leave the Merge with the Original checkbox in the Mirror Options dialog box unchecked. Instead, I merge the vertices after the mirror function. To do so, switch to Front view and select the vertices along the center line. Then merge them as shown in Figure 4.107.

Watch the vertex count and make sure it is reduced by half in the merge function. If it isn't, you most likely don't have all the vertices directly next to its mirrored counterpart. You can zoom in to find any problems; then snap them together and merge again.

You need to do just one more thing to finish the character: Open the Normals menu and select Soften Edge, as shown in Figure 4.108.

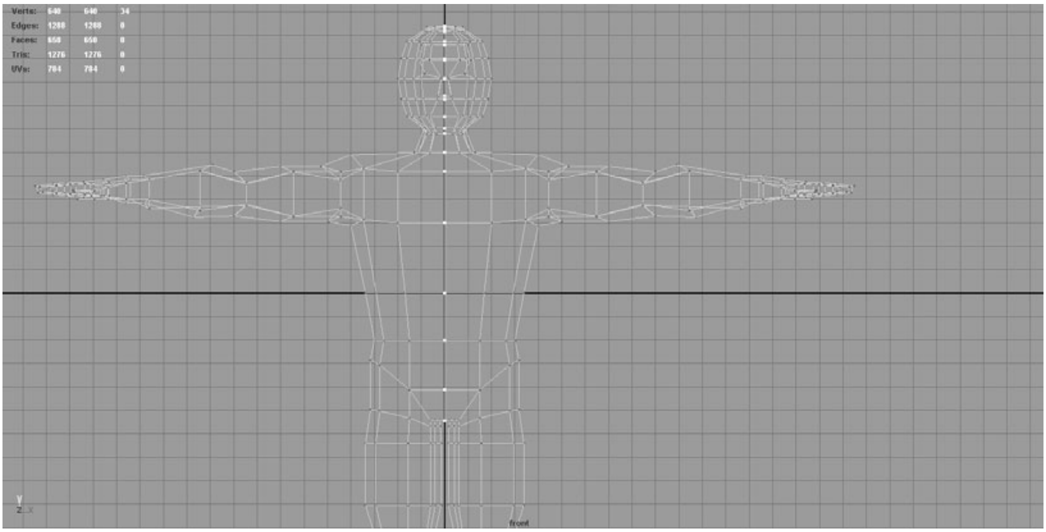


Figure 4.107
Merge the vertices along the center line.

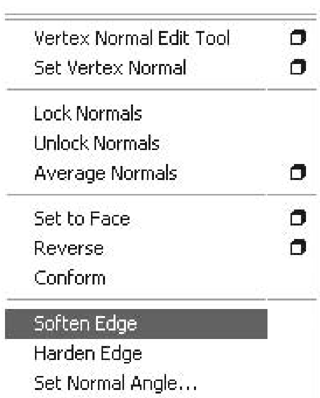


Figure 4.108
Soften the character model.

You have completed a simple game character model. It isn't highly detailed, but it is a good start in modeling characters. In the next chapter, you will create a more detailed game character.

Summary

You have just toured the interface and menus in Maya and should now have at least a cursory knowledge of the program's tools and features. The topics covered include the following:

- An overview of the Maya interface
- Building and texturing a ball
- A look at the menu system
- A quick explanation of the menu items
- An explanation of Maya's primitives
- Some of the more common polygon modeling functions
- A simple example of building a game character

As you continue on through the next few chapters, you will gain an even broader knowledge of the program. Don't be afraid to experiment and try a few models on your own.

This page intentionally left blank

CHAPTER 5

Character Modeling 2

The last chapter gave a very simple example of creating a game character to familiarize you with Maya and the tools used for creating characters. This chapter takes a more serious look at modeling a game character by showing the steps used in building one from start to finish. Specifically, it covers the following:

- Setting up the template
- Modeling the face
- Modeling the head
- Modeling the breast plate
- Modeling the arm
- Modeling the hand
- Modeling the leg
- Modeling the foot
- Finishing the model
- Mirroring the model

Setting Up the Template

Before you start to model your character you need to create a template to use as a guide. Figure 5.1 shows the template drawing of the character that you will create in this chapter. Notice that the template includes four drawings, each proportional

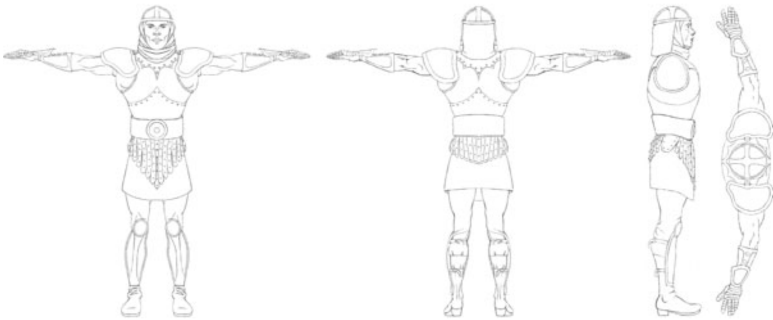


Figure 5.1
The template has four views of the game character.

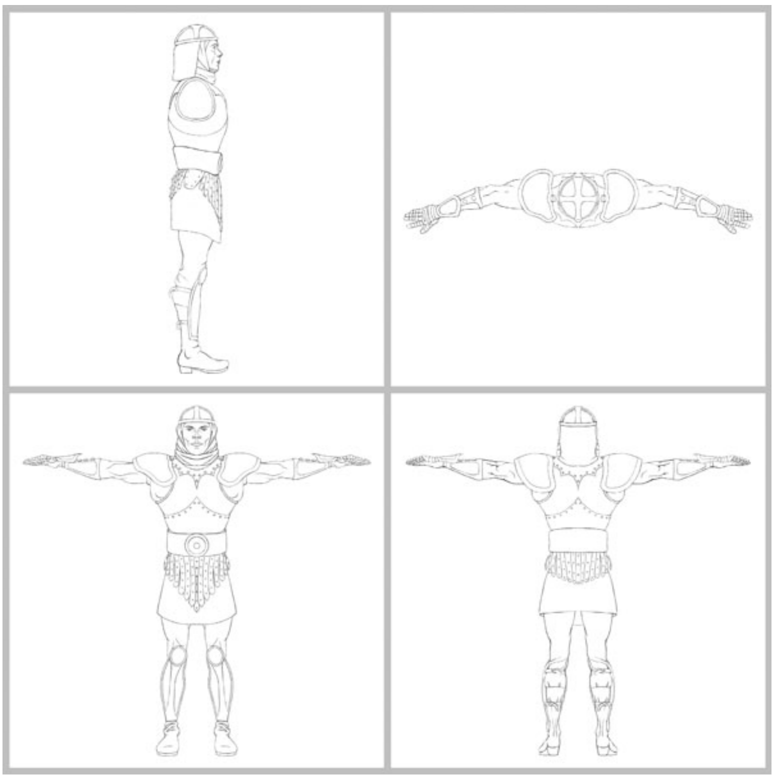


Figure 5.2
Center the drawing on four separate files.

to each other. (For more information on creating character templates, refer to Chapter 3, "Character Illustrations and Templates.")

To make the template, start by centering each drawing on a separate file, making sure that the dimensions of each file match, as shown in Figure 5.2. My drawings are centered on a 1600 × 1600 pixel bitmap.

Now you are ready to create the character template. Open Maya and do the following:

1. Go to the Hypershade editor by opening the Window menu, choosing Rendering Editors, and selecting Hypershade.
2. You'll need to create some new materials with your drawing. I like to use Lambert materials for templates because they have a matte finish, whereas Blinn and Phong materials are shiny, making them hard to see. To create a new Lambert material, open the Create menu in the Hypershade window, choose Material, and select Lambert. Figure 5.3 shows the result.
3. Launch the Attribute editor and load the drawing of the front view into the Color attribute, as shown in Figure 5.4. The color attribute is the top one listed in the editor. Rename the material Front.
4. Repeat step 3 for the other three drawings so that you have a front, side, back, and top material, as shown in Figure 5.5.

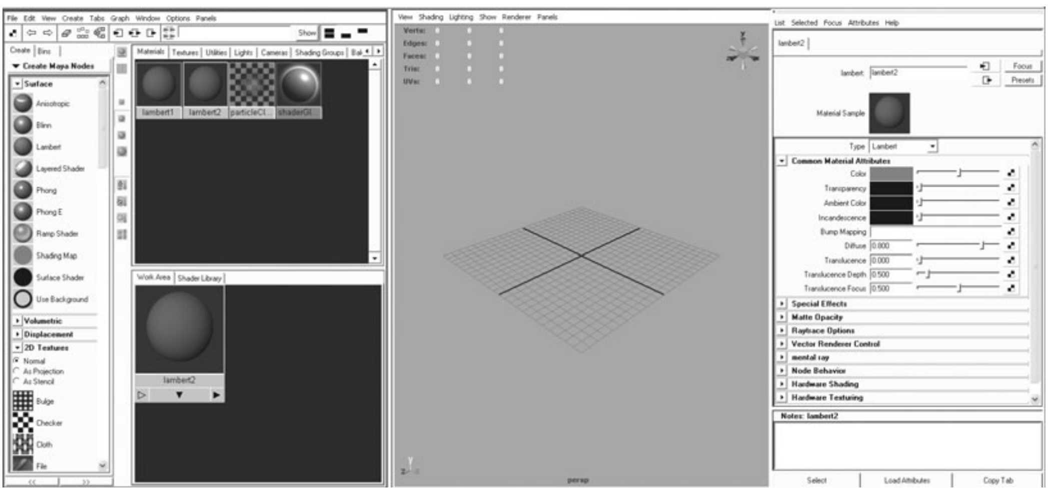


Figure 5.3
Create a new Lambert material for the template.

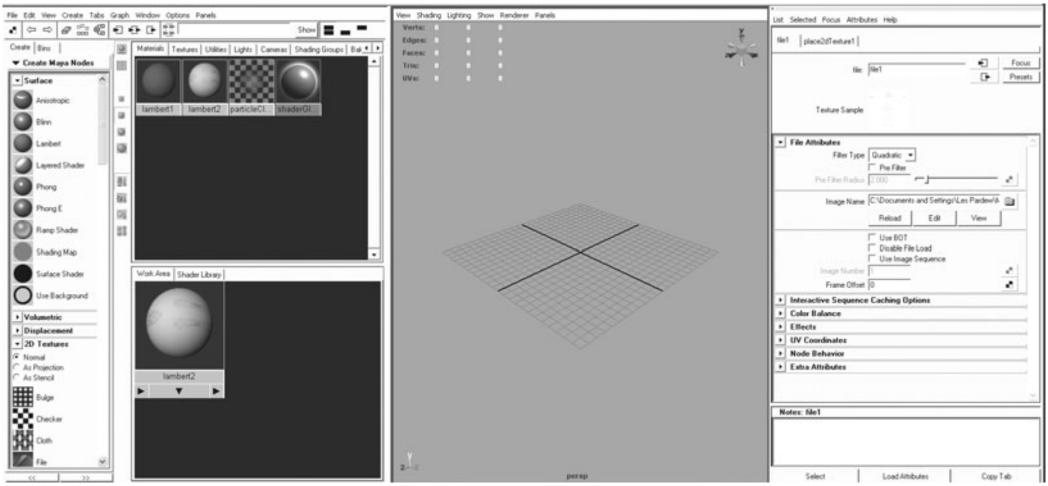


Figure 5.4
Load the front drawing into the Color attribute in the Attribute editor.

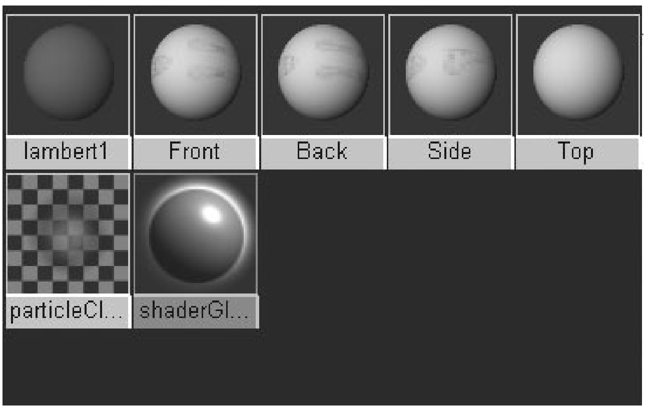


Figure 5.5
Create a material for each drawing.

5. Create a square polygon plane that is 16 × 16 units, as shown in Figure 5.6. Make it face forward in the front view by creating it in the z axis.

6. Translate the new polygon to .01 in the z axis.

7. Repeat step 5 to create another square polygon plane.

8. Rotate the new polygon plane 180 degrees so that it is facing the opposite direction from the first plane.

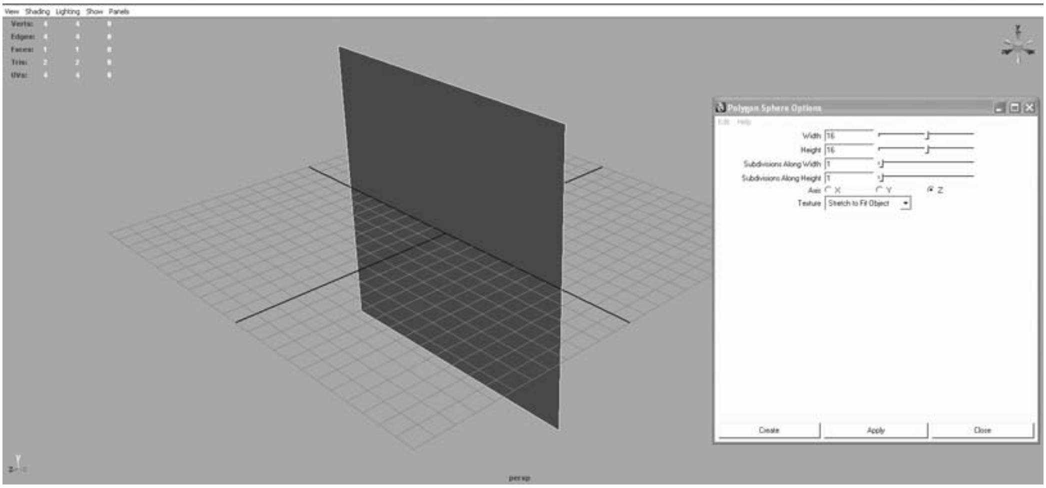


Figure 5.6
Create a square polygon plane.

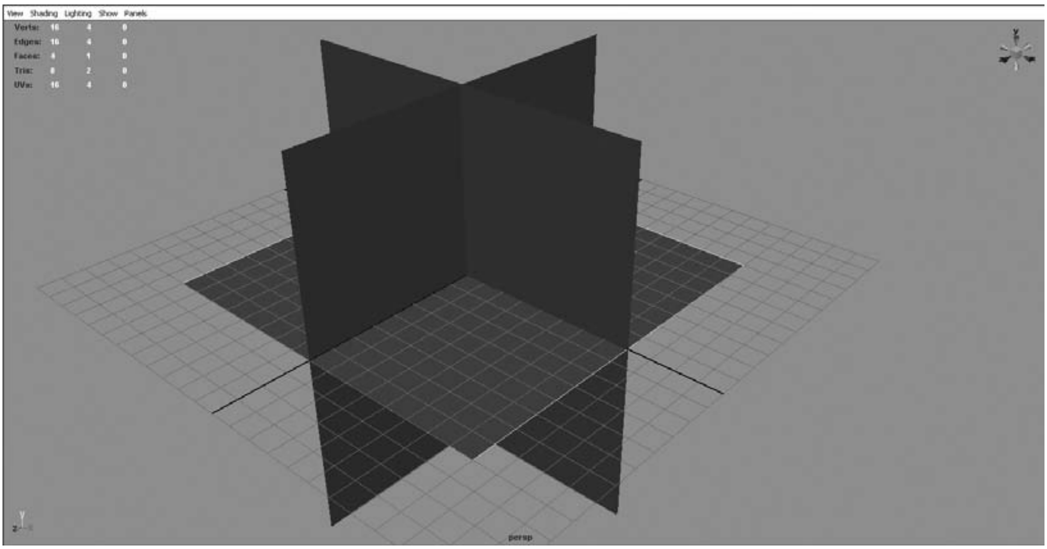


Figure 5.7
Create four polygons for the character template.

9. Move the new polygon to −.01 in the z axis. (Moving the polygons away from the center line ensures that the front and back views will not bleed into each other.)
10. Make one polygon plane in the x axis and another polygon plane in the y axis for the side and top views. Figure 5.7 shows how the polygons should look.

11. Assign the file that contains the front view to the front-facing polygon, as shown in Figure 5.8. Notice that the character falls directly in the middle where the polygon is bisected by another polygon.

12. Apply the side material to the side-facing polygon.

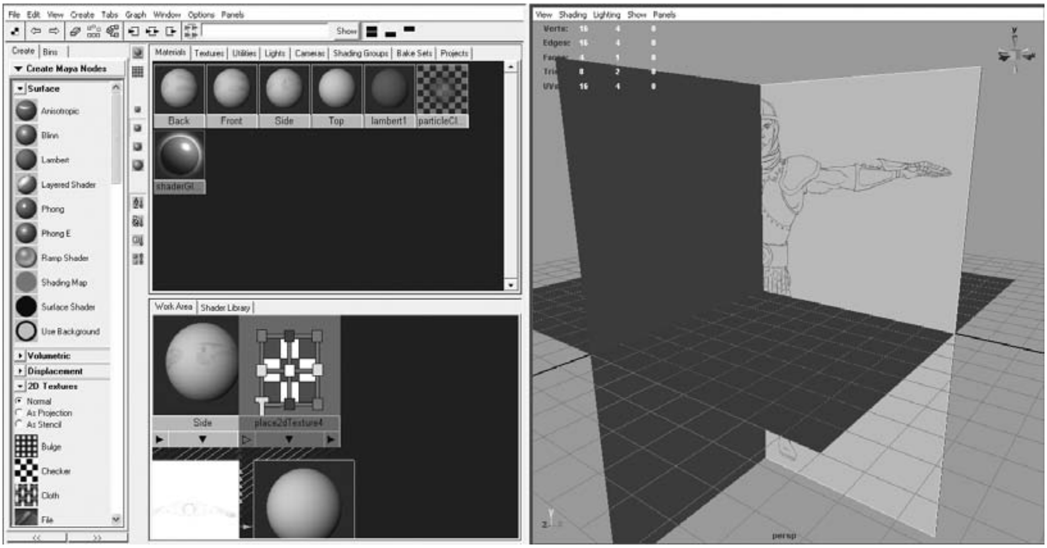


Figure 5.8
Apply the front material to the front-facing polygon.

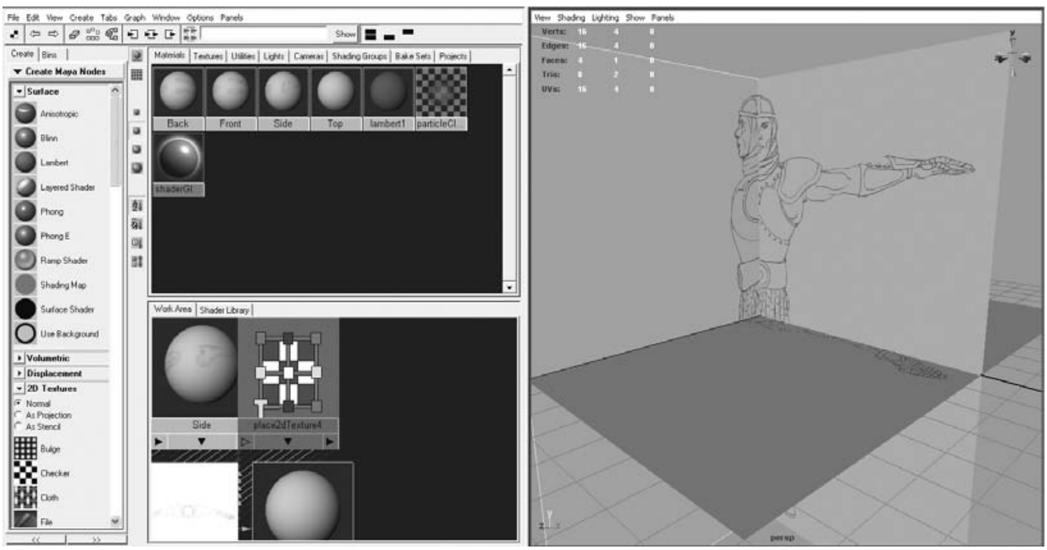


Figure 5.9
Each drawing is placed on the template, facing in the appropriate direction.

13. Notice that the drawing is facing the wrong direction. One way to fix this problem is to flip the drawing in a paint program and then reload it into the material. Another way is to rotate the polygon 180 degrees in Maya in the y axis. (If you rotate the drawing in Maya, you will also need to reverse the direction of the normal. To do so, open the Edit Polygon menu, choose Normals, and select Reverse Normal.)
14. Apply the top material to the polygon facing up in the y axis.
15. Apply the back material to the polygon facing back, or away from the front view. Figure 5.9 shows the finished template.

Modeling the Face

Although you can start modeling a character from almost any part of the body, a good place to start is the face. That's because the character's face will usually be the focal point for the figure. In the front view, move in close the character's face, as shown in Figure 5.10.

The Eye

For the character's face to animate well, the polygons of the face need to be created in concentric rings emanating from the eye and the mouth. This will help

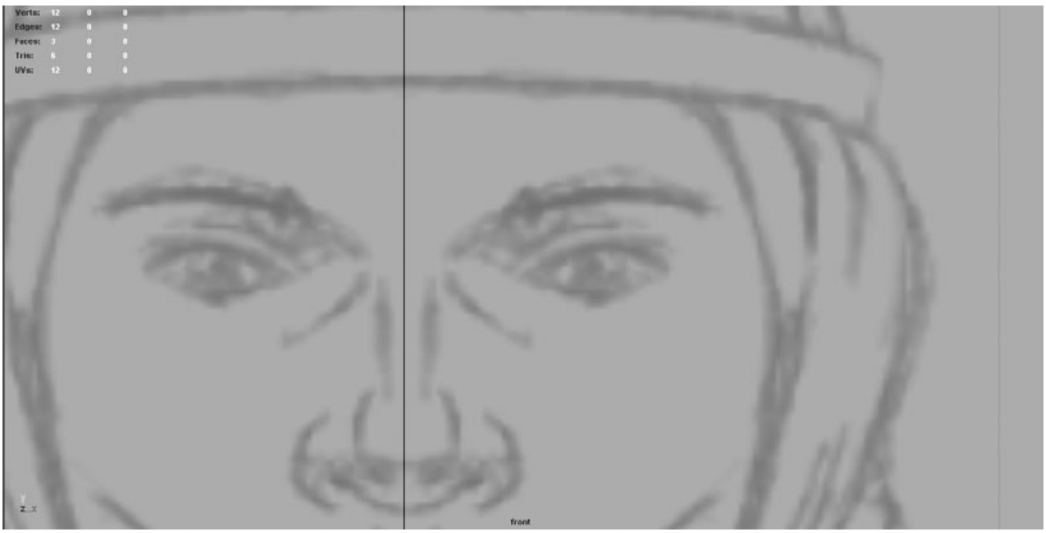


Figure 5.10
Begin modeling the character by moving in close to the face.

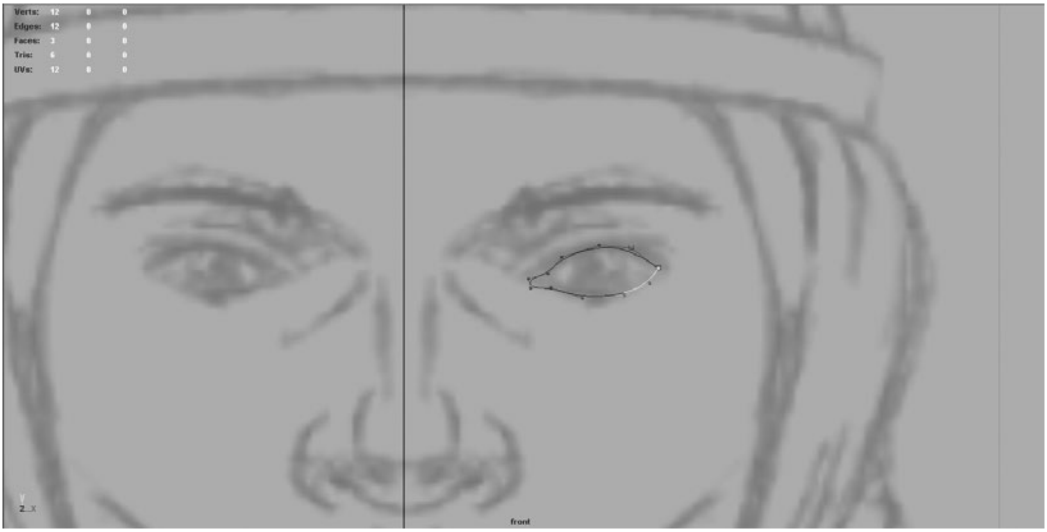


Figure 5.11
Draw a CV curve around the eye.

to reduce unwanted deformations when creating facial animations. Start building the character in the area around the eye.

1. First, create a CV curve around the eye, as shown in Figure 5.11. To begin, open the Create menu and choose CV Curve.
2. Using the CV Curve tool, click the outside corner of the eye, and trace the curve of the upper eyelid, around the tear duct, around the lower eyelid, and back to the point of origin, clicking as you go. When you have traced the entire eye, press Enter.

Note

When drawing CV curves, it is helpful to be in one of the Orthographic views. In these views, the curves are placed on the center line, making them easy to find in 3D space.

3. Move the finished CV curve forward so it lines up with the character's eye in the side view.
4. Right-click the CV curve and choose Control Vertex in the menu that appears to change the selection mode to Control Vertices.
5. Use the control vertices follow the curve of the eyelids around the eyeball in order to round the shape of the eye, as shown in Figure 5.12.

6. Return to Object Selection mode by clicking the button in the status line shown in Figure 5.13 and selecting the CV curve.

7. Press Ctrl+D to duplicate the curve, and then move the curve forward as shown in Figure 5.14. You will be lofting between the curves later, so you need the curves to have the exact same number of CVs. Duplicating the curve is an easy way to match curves for lofting. This new CV curve should follow the curve of the outer ridge of the eyelid.

8. Scale the CV curve and adjust the individual control vertices to follow the shape of the eyelid.

9. Duplicate the CV Curve and scale it outward to follow the area around the eye. Adjust individual Control Vertices where needed. Refer to Figure 5.15 to see how the curve should look from the front view. There should now be three curves.

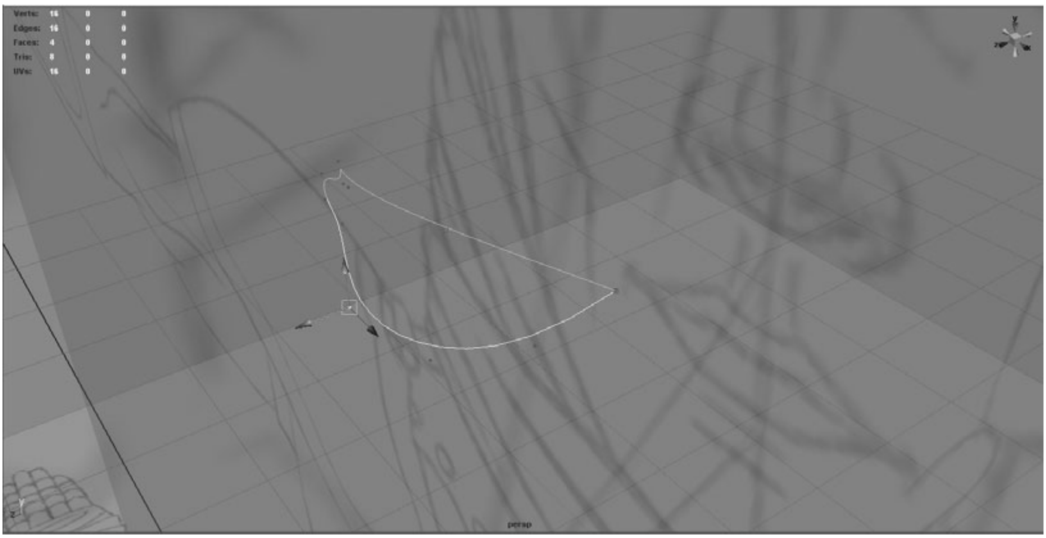


Figure 5.12
Adjust the shape of the CV curve to follow the curve of the eye.

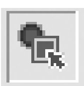


Figure 5.13
Change the selection mode to object.

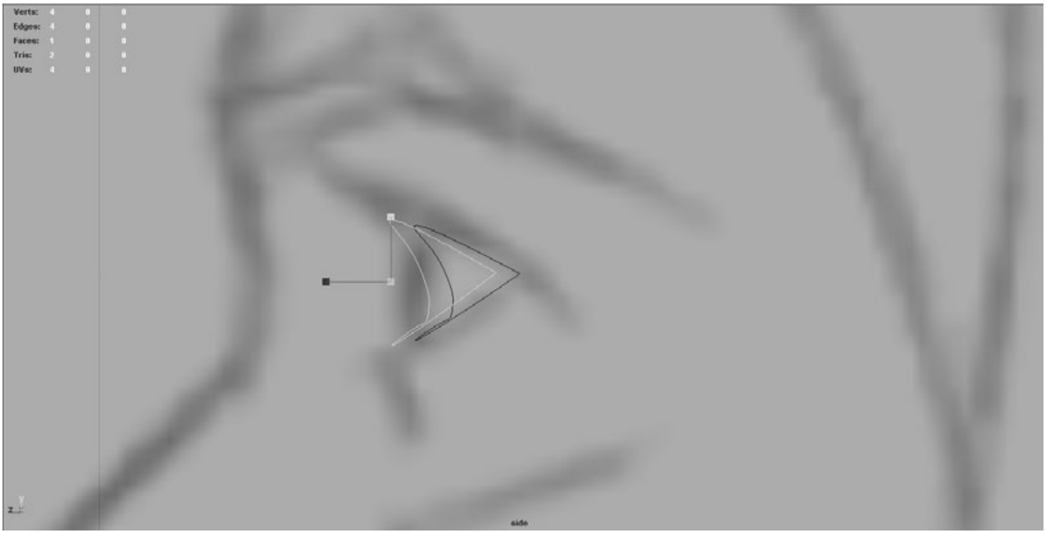


Figure 5.14
Duplicate the CV curve and move the new curve forward.

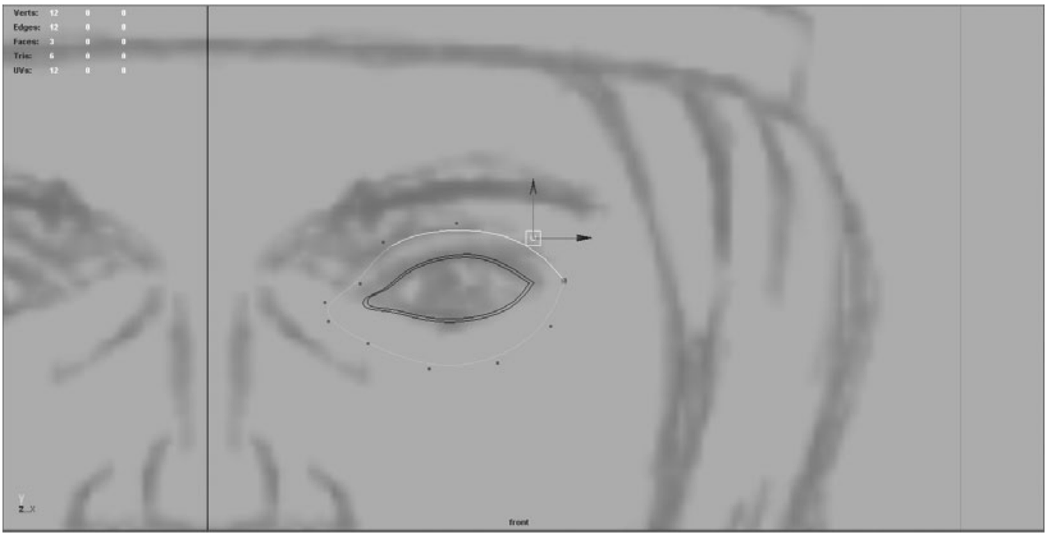


Figure 5.15
Duplicate the CV curve and scale it to form another ring.

10. Now that you have a couple of CV curves, you can loft polygons between them. *Lofting* is the process of spanning the area between curves with polygons or some other surface. To access the Loft tool, open the Surfaces menu set, choose Surfaces, and choose Loft; the Loft Options dialog box opens.

11. Change the Parameterizations settings to Uniform and check the Auto Reverse checkbox.
12. Change the Surface Degree setting to Cubic.
13. Change the Selection Spans setting to 1.
14. Choose the Partial option button next to Curve Range.
15. Choose the Polygons option button next to Output Geometry.
16. Uncheck the Match Render Tessellation checkbox.
17. Choose the Quads option button next to Type.
18. Change the Tessellation Method setting to General.
19. Choose Per Surf # of Iso Params from the U Type drop-down list.
20. Choose Per Surf # of Iso Params from the V Type drop-down list.
21. Type `16` in the Number U field.
22. Type `1` in the Number V field.
23. Leave the rest of the options as is. Figure 5.16 shows how the Loft Options dialog box should look.
24. In the work area, select the outer CV curve, hold down your Shift key, and then select the next curve.
25. Loft polygons between the two CV curves (refer to Figure 5.16) by clicking the Apply button in the Loft Options dialog box.

Note

The advantage of using CV curves over just creating polygons is that you can use the curves to make fine adjustments to the polygons even after they have been lofted—as long as you leave History feature on. (The History feature is enabled by default.)

26. Use the same settings to loft between the two inner CV curves. Notice that the polygons line up perfectly between the curves.
27. Duplicate the outer curve and use it to create another concentric ring around the eye, as shown in Figure 5.17.

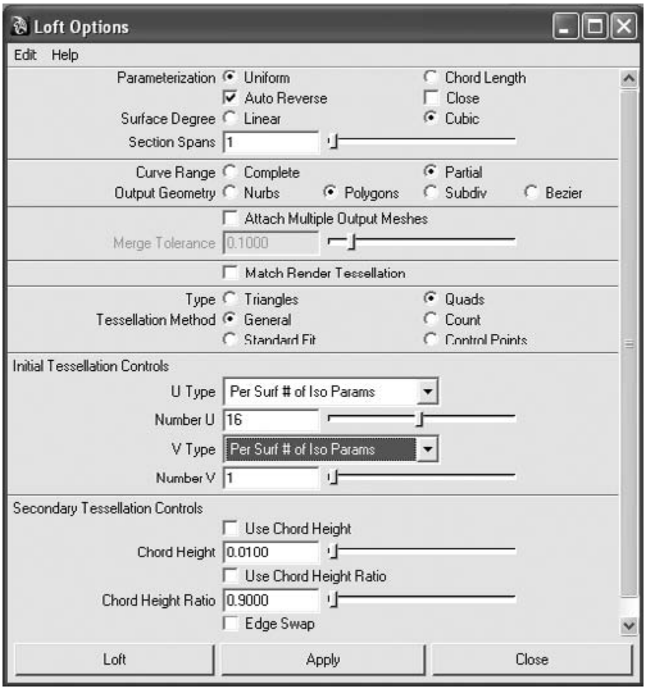


Figure 5.16
Establish your settings in the Loft Options dialog box.

The Mouth

Like the eye, the mouth should be modeled with concentric rings of polygons. In this example you will use CV curves in a similar manner to what you just did with the eye.

1. Create a CV curve around the lips, as shown in Figure 5.18.
2. Unlike with the eye, you need to join the ends of the CV curve around the mouth so you will better be able to build the cheeks. To do so, open the Edit Curves menu and choose Open/Close Curves. In the Open/Close Curves Option dialog box, select the Blend option button, and change the Blend Bias setting to 0.5. Check the Insert Knot check box and change the Insert Parameter setting to 0.1, as shown in Figure 5.19.
3. Duplicate the curve and scale it down to form the inside of the lips where they meet. Shape the curve as needed. See Figure 5.20 for an example.
4. Duplicate the curve again and modify the new curve to follow the curve of the lips between the other two curves.

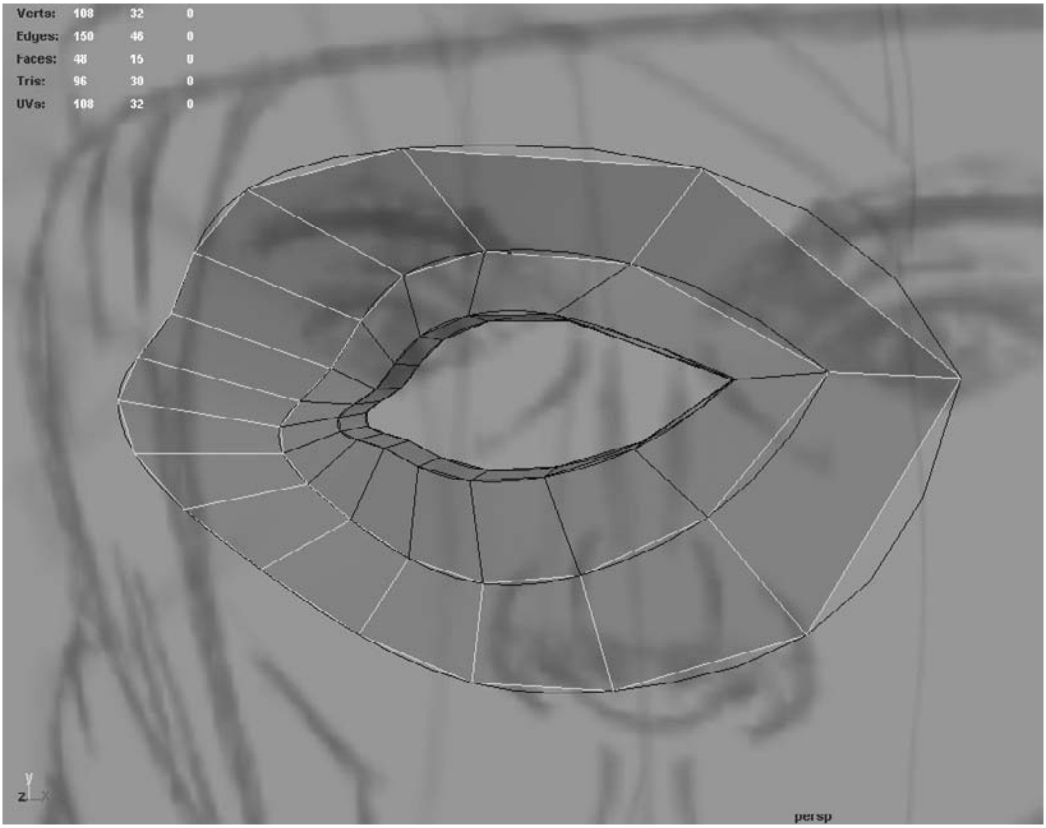


Figure 5.17
Create another concentric curve around the eye.

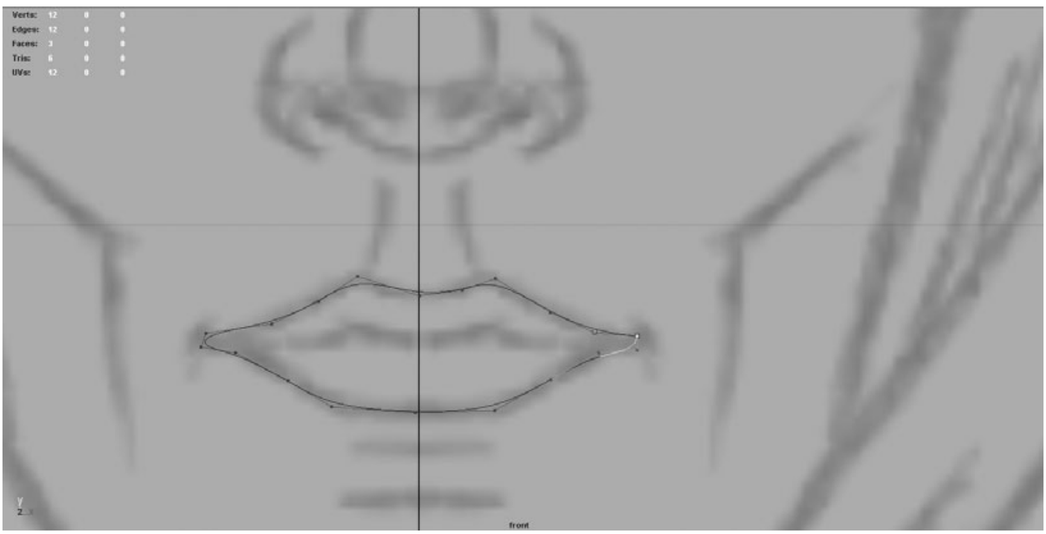


Figure 5.18
Draw a CV curve around the lips.

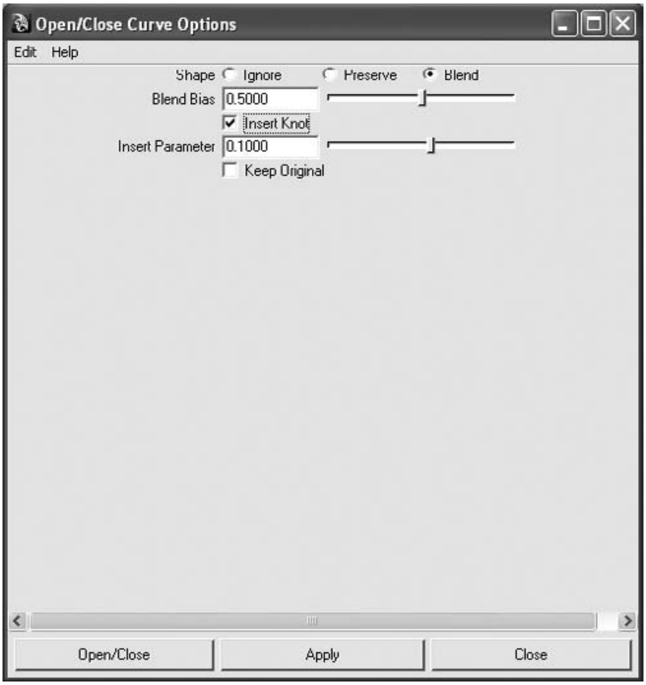


Figure 5.19
Close the ends of the CV curve.

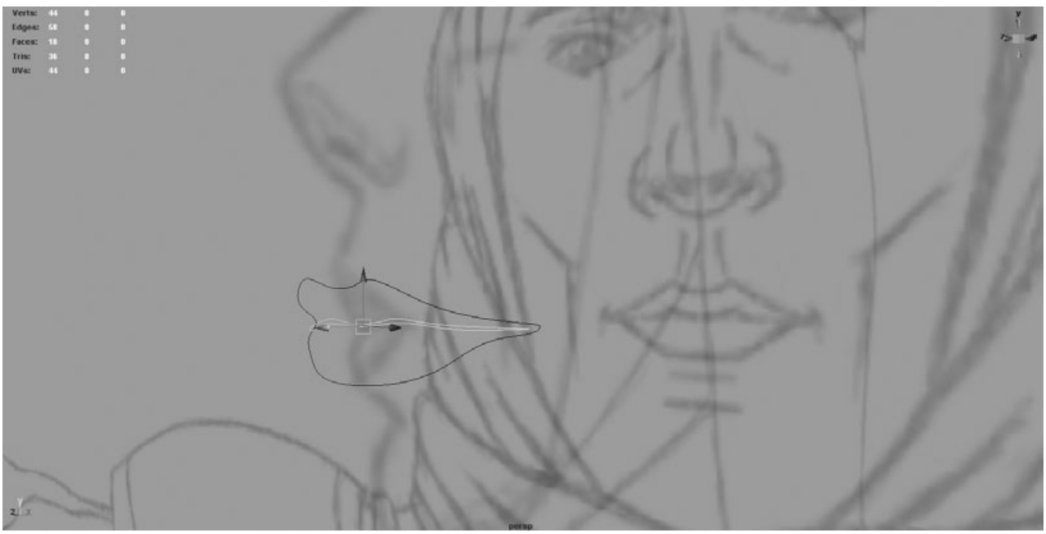


Figure 5.20
Duplicate the curve and use the duplicate to form the edge where the lips meet.

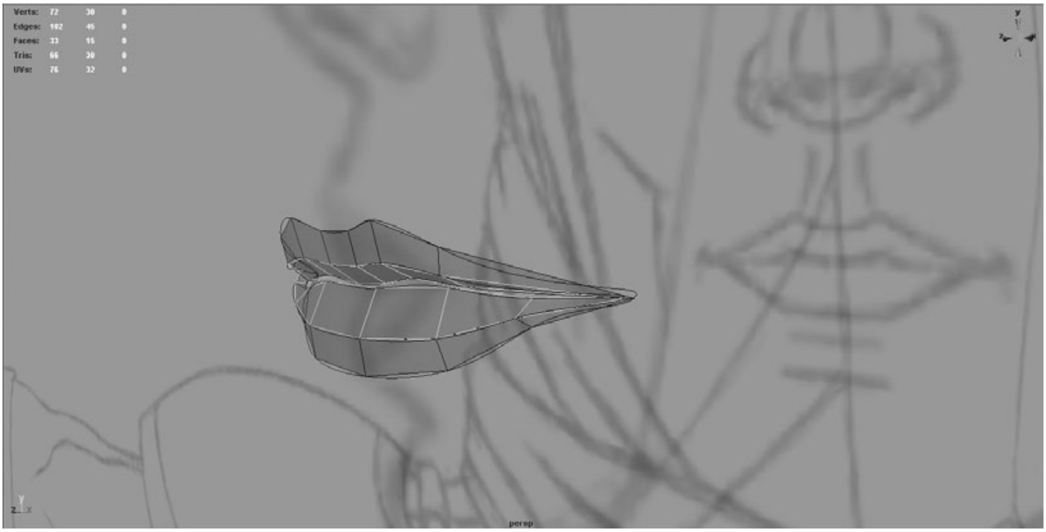


Figure 5.21
Loft polygons for the lips.

5. Loft between the curves using the same settings you used for the eyes to form the polygons of the lips, as shown in Figure 5.21.
6. Duplicate the curve again and scale it outward from the lips.
7. Loft polygons between the new curve and one around the lips, as shown in Figure 5.22. Adjust the control vertices to follow the shape of the mouth.
8. Duplicate the curve one more time and loft another ring of polygons around the mouth extending up to the base of the nose, as shown in Figure 5.23.

The mouth and eye are now complete; the next step is to create the character's nose and the rest of his face. Before you move on, check the shape of the eye and mouth in the Perspective view and make any necessary adjustments to make them look better. Using the control vertices should make this process easier.

The Face

The next step in building the character model is to connect the eye with the mouth, first by building the nose and then the cheeks, chin, and forehead.

1. Create another ring of polygons around the eye, this time sweeping up to the bridge of the nose as shown in Figure 5.24.

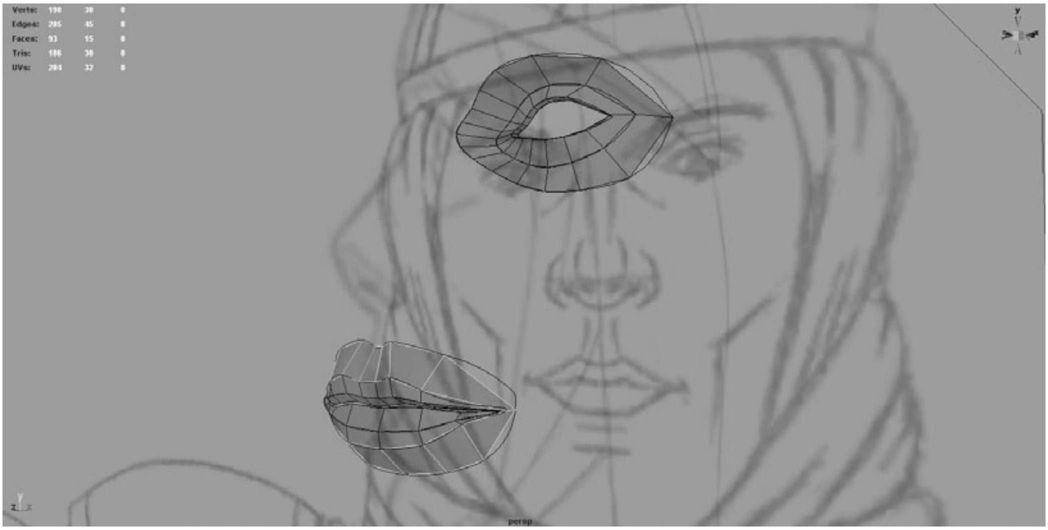


Figure 5.22
Build outward from the mouth.

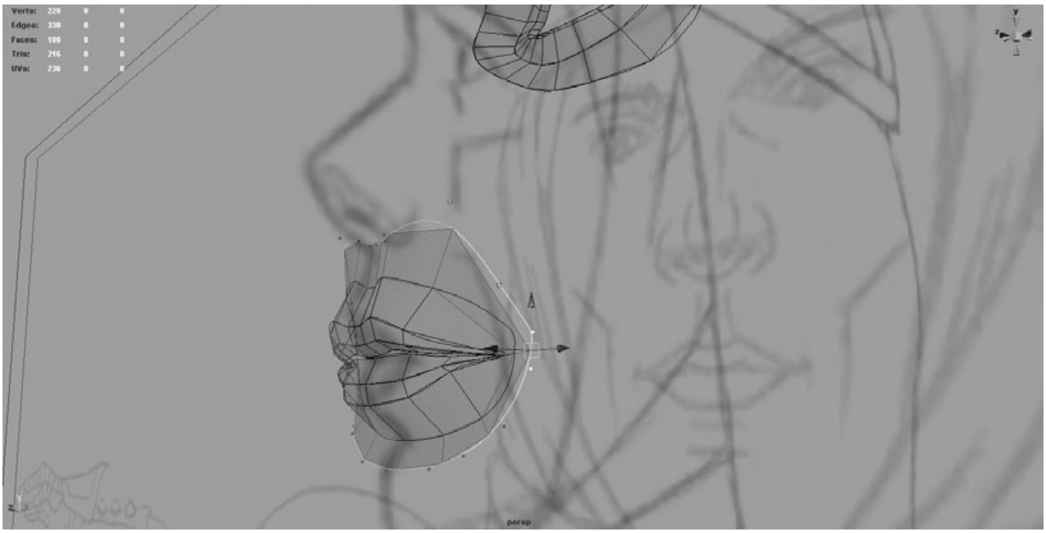


Figure 5.23
Loft another ring of polygons surrounding the mouth.

2. In the front view, select the edges of the three polygons along the bridge of the nose and extrude them to the center line, as shown in Figure 5.25.
3. Extrude the edge of the lower bridge polygon down the length of the nose to the polygons of the mouth, as shown in Figure 5.26.

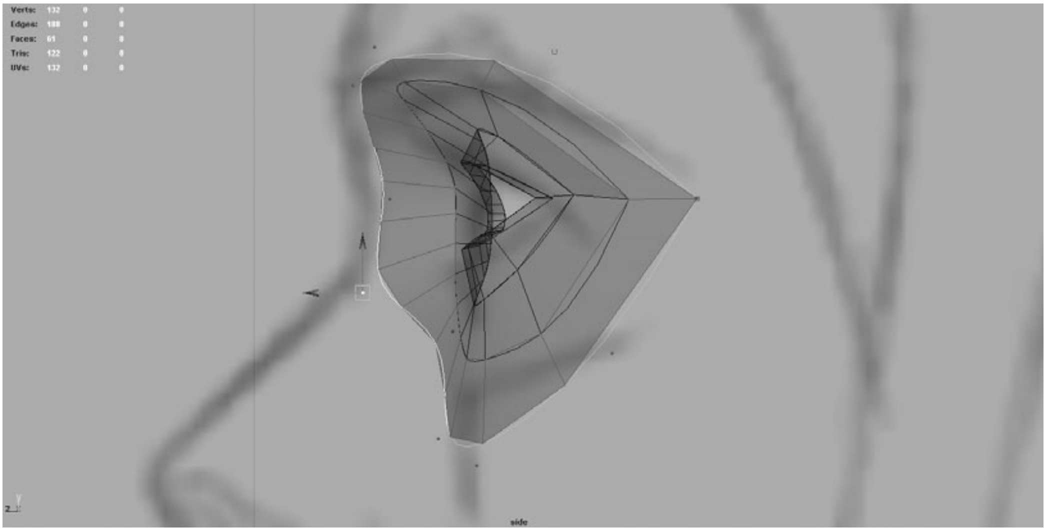


Figure 5.24
Create another ring of polygons around the eye.

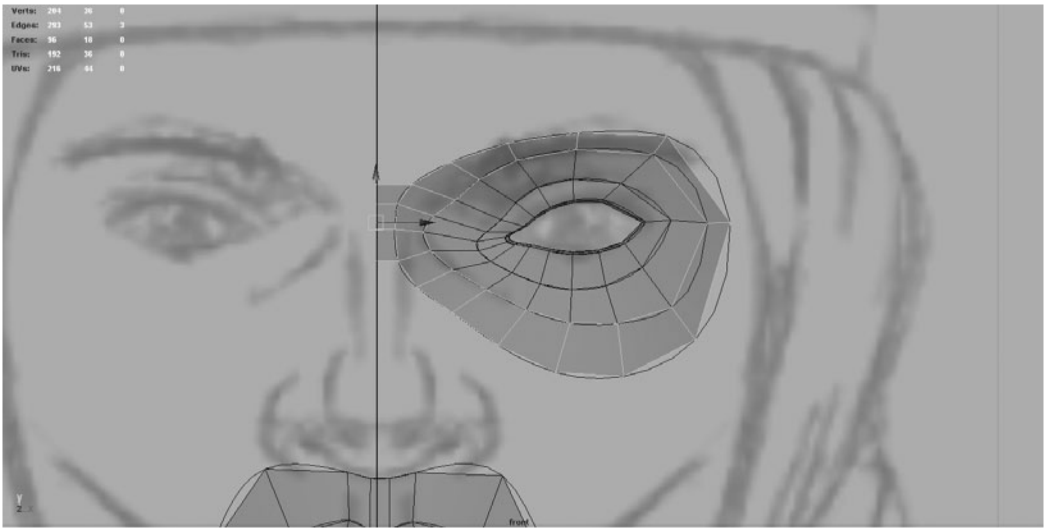


Figure 5.25
Extrude the polygons at the bridge of the nose.

4. Go to the side view and use the Move tool to move the vertices of the extruded polygons to follow the shape of the nose, as shown in Figure 5.27.

5. Now extrude the edges of the nose sideways along the nose to build the nose. After each extrusion, use the marking menu to change the selection mode to Vertex and move the vertices to create the nose. Snap the tops of the

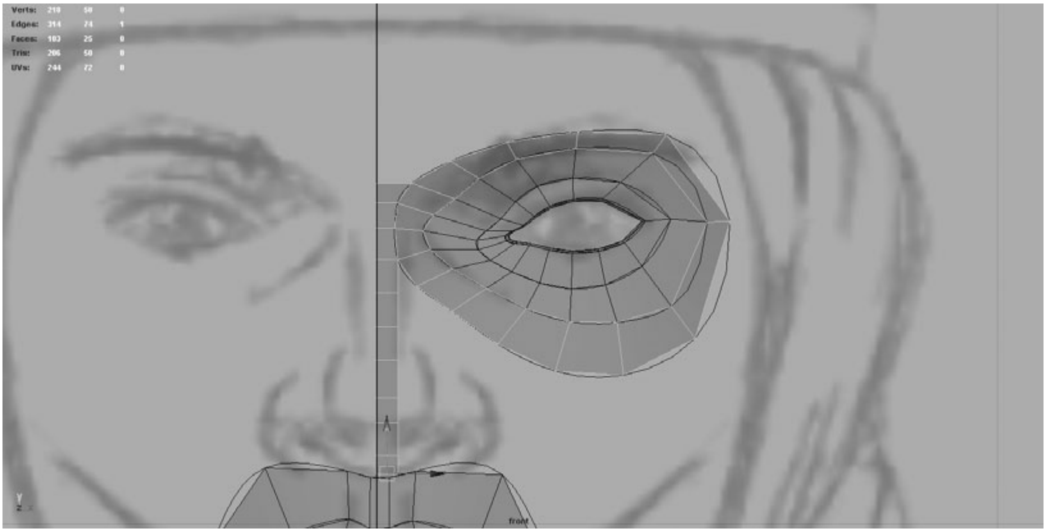


Figure 5.26
Extrude the edges from the bridge of the nose to the mouth.

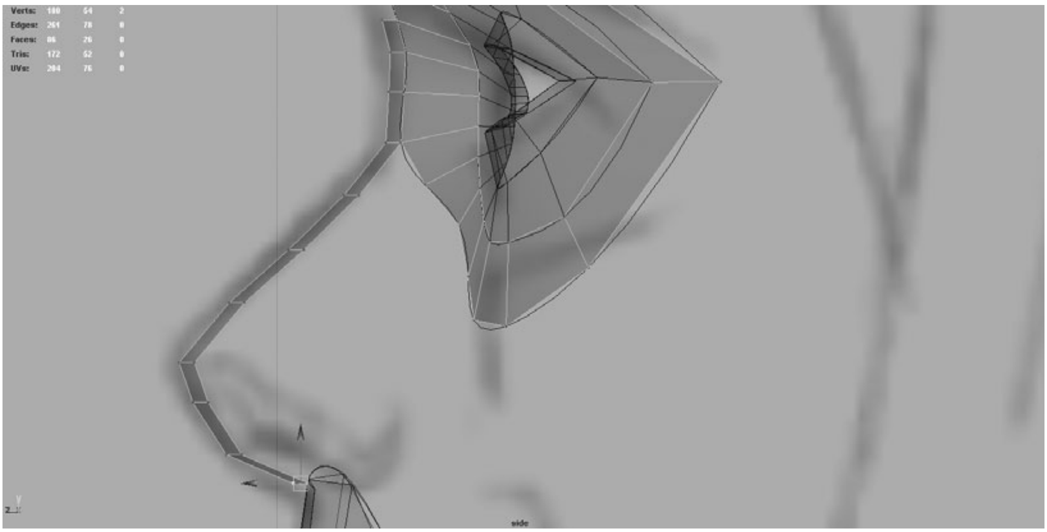


Figure 5.27
Move the vertices to follow the template of the nose.

extrusion to follow the lower ring of the eye. Use Figure 5.28 as a guide for how the nose should look.

6. So far, each ring of polygons around the eye and the mouth are separate objects. To make one continuous mesh, you need to combine them into a single object and then merge the objects together. Select all the polygon

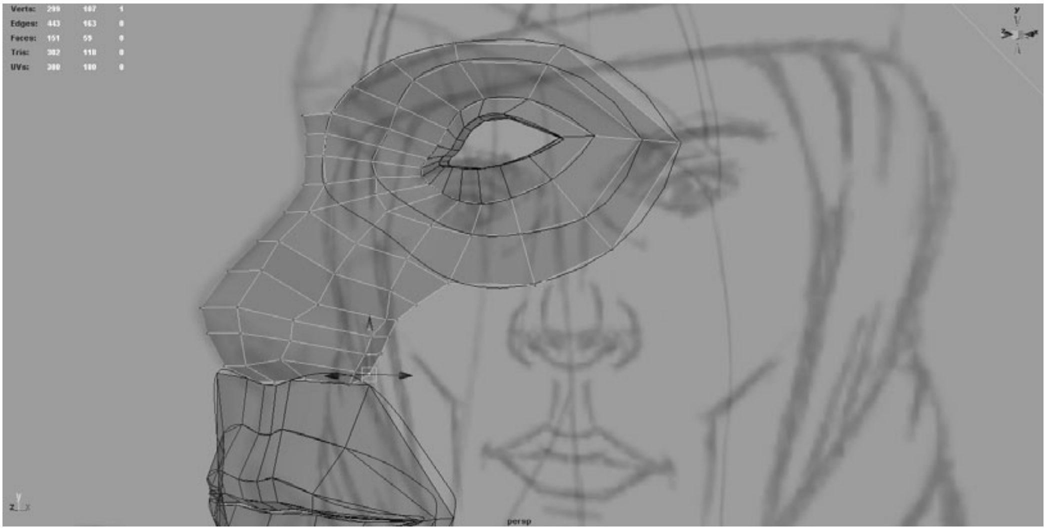


Figure 5.28
Build the nose by extruding edges and moving vertices.

objects and combine them by selecting the Combine function from the Mesh menu of the Polygons menu set, as shown in Figure 5.29.

7. Before you go any farther, you need to clean up a couple of things. First go to the front view and snap the vertices of the mouth to follow the center line, as shown in Figure 5.30. To snap one vertex to another, select the vertex you want to move and, using the Move tool, move the vertex toward the center line to which you want it to snap while holding down the X key on the keyboard.
8. Next, delete the polygons on the left side of the screen from the center line. (Because you will be using the mirror technique on the model, you only need to create half of the face.)
9. You no longer need the CV curves, so those can be hidden. To hide them, select all the curves and press the Ctrl+H keyboard shortcut.
10. Open the Edit Mesh menu and choose Merge Options to merge the vertices in the model. This removes duplicate vertices and connects all the rings around the mouth and eye (see Figure 5.31).
11. Join the mouth with the rest of the model by snapping the vertices of the nose and mouth together and merging them. Use Figure 5.32 as a guide of how they should come together.

Figure 5.29
Combine all the polygon objects.

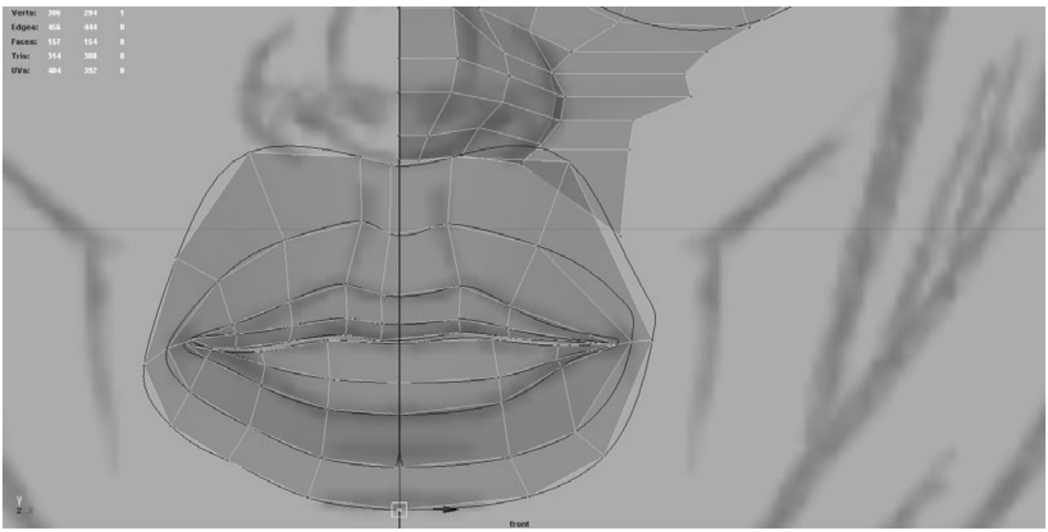


Figure 5.30
Snap the vertices of the mouth to the center line.

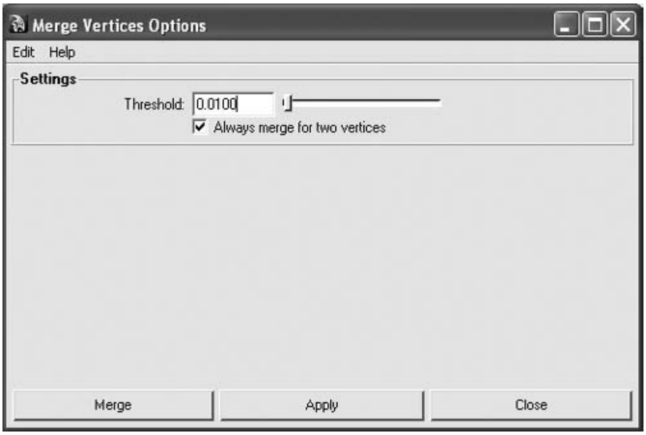


Figure 5.31
Merge the vertices of the model.

Figure 5.32
Join the mouth polygons with the nose polygons.

12. You have the main facial features, which means most of the hard work is done. All you need to do is extrude outward to the edges of the face. Start by extruding the edges along the brow to the bottom of the helmet, as shown in Figure 5.33. Make two extrusions.

13. Next, extrude the edge from the chin to the brow to the edge of the head wrap. Use Figure 5.34 as a reference for your extrusions.

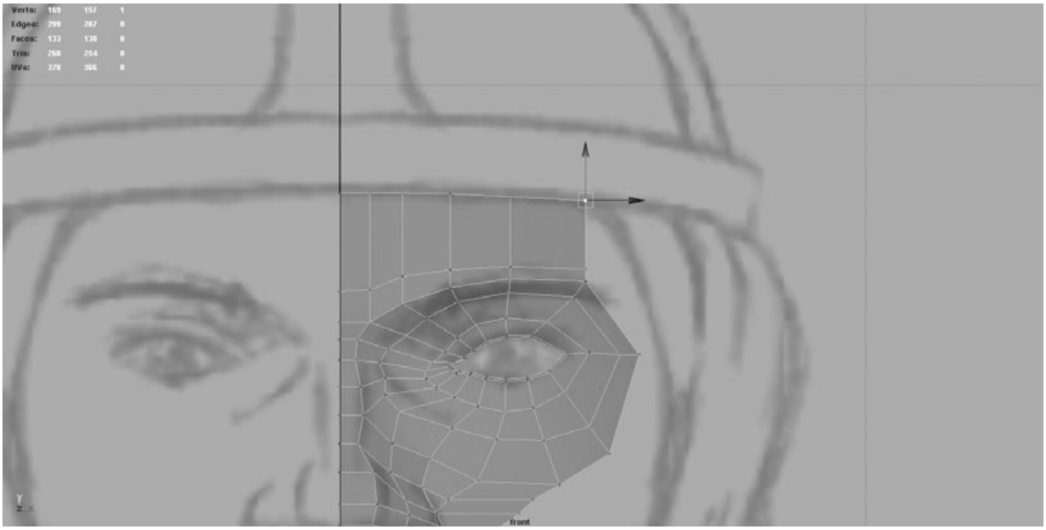


Figure 5.33
Extrude the edges along the brow to the bottom of the helmet.

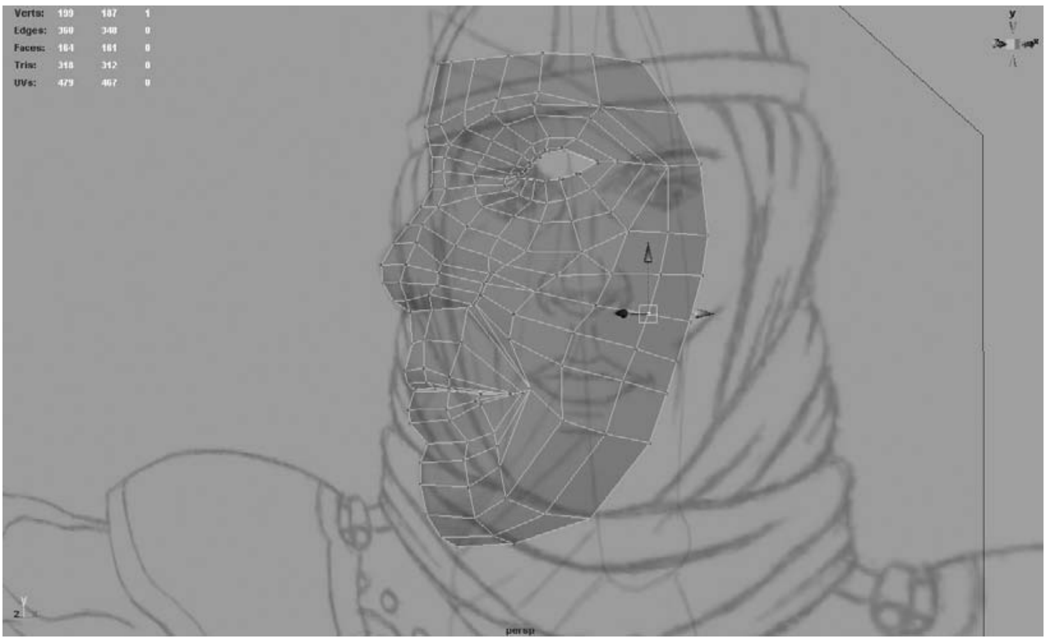


Figure 5.34
Extrude the edges from chin to brow to form the side of the face.

The face is now finished. To wrap things up, go back in to make sure the structure of the head and the facial features are correct. Adjust vertices where needed around the features and along the cheekbone and chin.

Modeling the Head

Other than the face, the rest of the character's head is made up of a head wrap and a metal helmet with chainmail hanging down the back. In this section you will build these elements of the model.

The Head Wrap

The head wrap of your warrior is used to protect his head and neck in battle. It is made of flexible yet tough material to help blunt attacks to that area while giving the character a greater freedom of movement than an enclosed helmet would. To build this part of the character's head, you will extend the model from the face.

1. Extrude the edges along the side of the face around to the center line, as shown in Figure 5.35. Adjust the vertices to form the shape of the wrap.

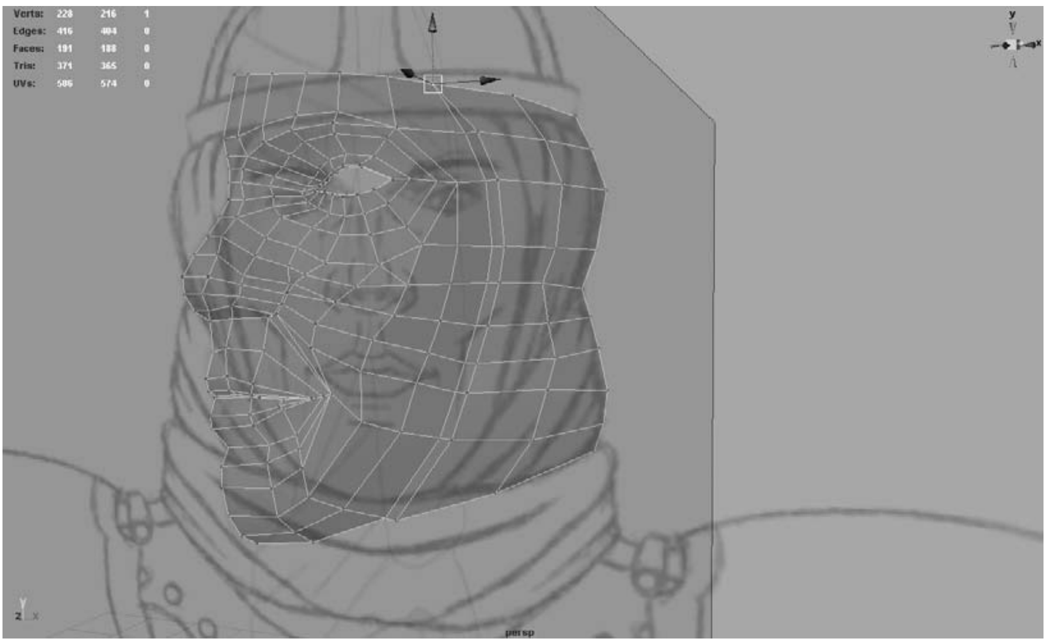


Figure 5.35
Extrude the edges from the side of the face to form the wrap.

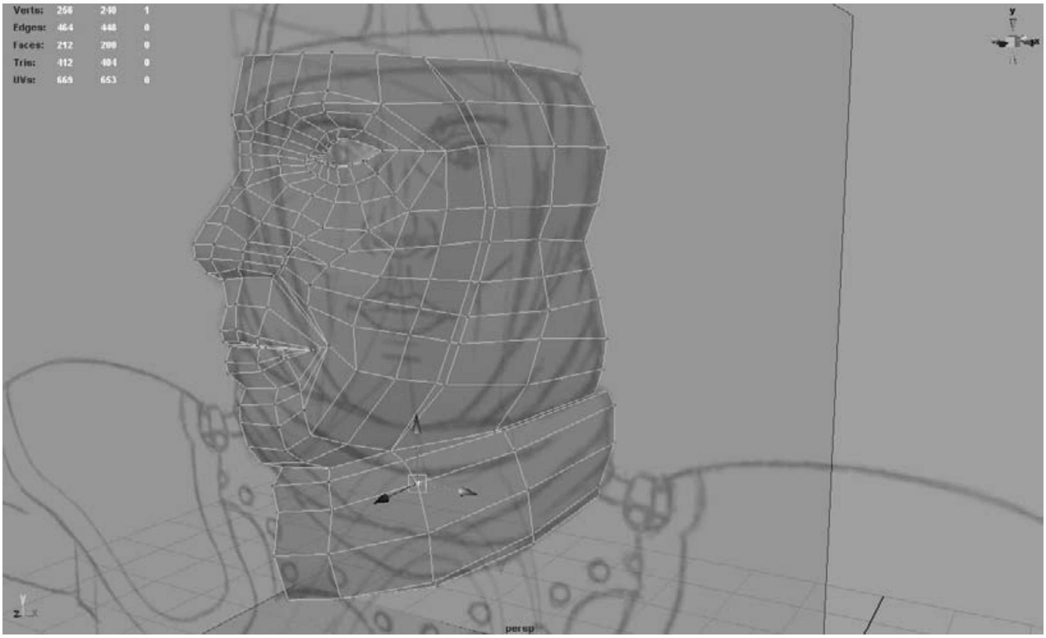


Figure 5.36
Extrude the edges down for the wrap around the neck.

Also notice that the vertices along the edge where the forehead, wrap, and helmet meet are merged.

2. Extrude the edges down to form the wrap around the character's neck, as shown in Figure 5.36.

3. Select all the edges along the back edge of the model. From the top view, extrude the edges and rotate them around the center axis, as shown in Figure 5.37. To move the pivot point of the Rotate tool to the center, click the tool and then press the Insert key. Press the Insert key again to bring the tool back and rotate the extruded edges.

4. Continue to extrude edges and rotate them around the center until you reach the center line at the back of the head, as shown in Figure 5.38.

5. In Vertex Selection mode, move the vertices of the back of the head to follow the template in the top view, as shown in Figure 5.39.

6. From the side view, move the vertices to fit the template (see Figure 5.40). Remember that the wrap needs to fit underneath the chainmail.

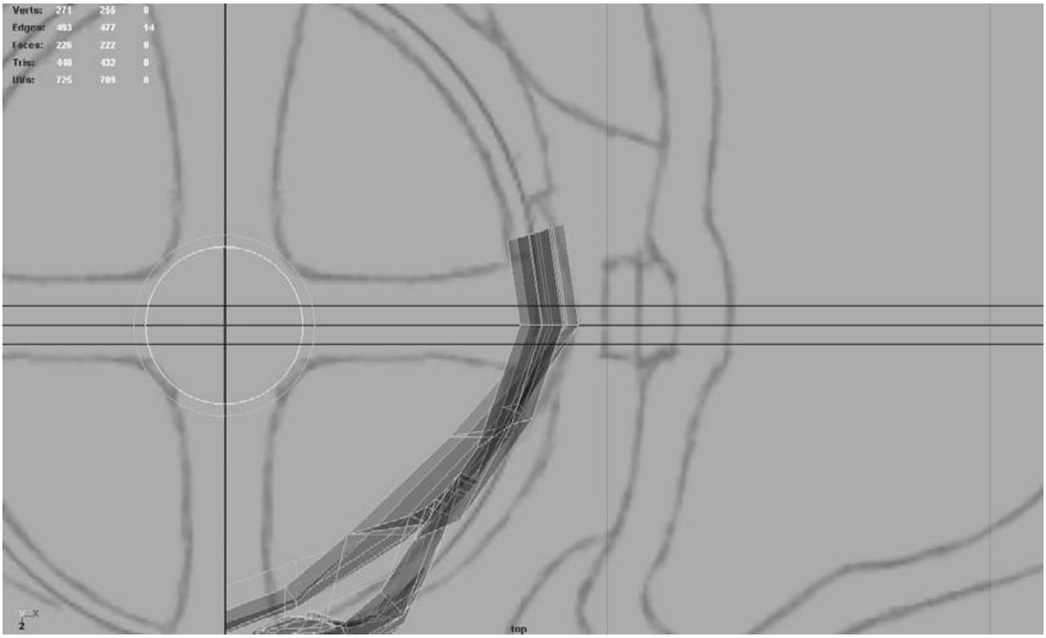


Figure 5.37
Extrude and rotate the back edge of the model around the center.

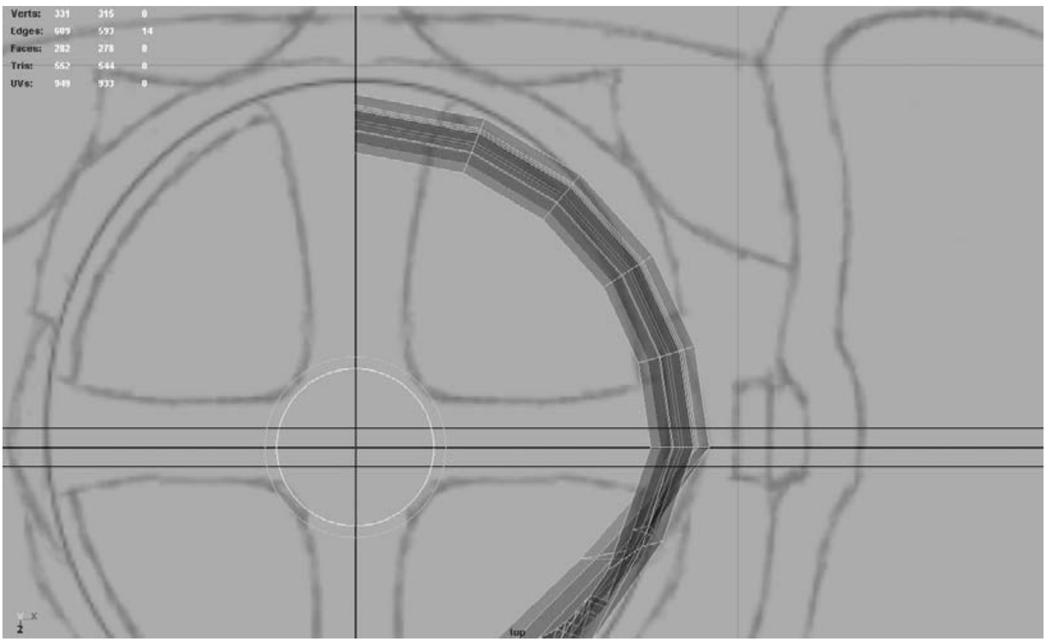


Figure 5.38
Continue extruding and rotating to the back center line.

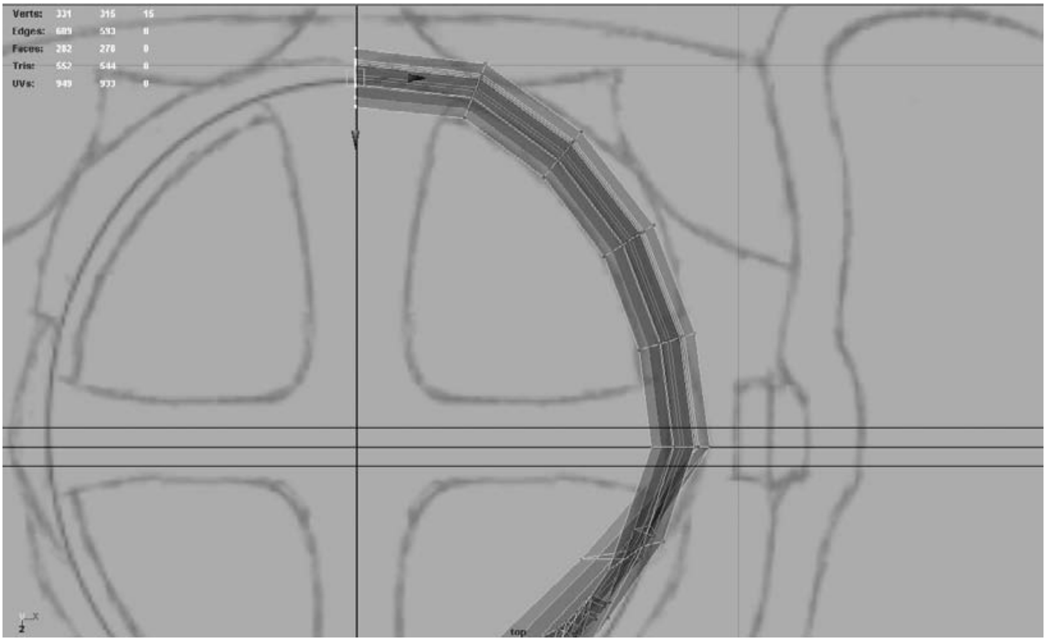


Figure 5.39
Move the vertices of the wrap to match the template.

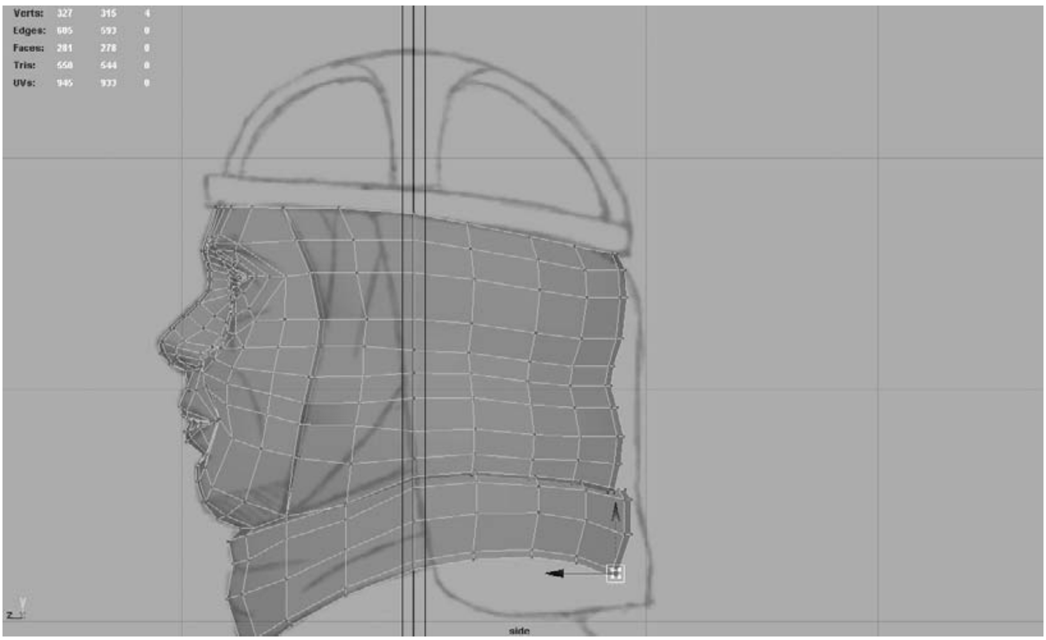


Figure 5.40
Follow the template for moving vertices in the side view.

Although the player may see parts of the wrap around the back of the character's head, the chances are slim because it will be covered by the chainmail. For this reason, there doesn't need to be a lot of detail there.

The Helmet

Rather than extruding upward for the helmet, you will create a separate polygon primitive and use it to build the helmet.

1. Create a polygon sphere with a radius of 1, with 26 subdivisions around the axis, and with eight subdivisions along the height, as shown in Figure 5.41.
2. Delete the bottom half of the sphere and move it to the top of the character's head. Rotate the sphere 6.92 degrees in the y axis to line the edges up with the center line. From the top view, the sphere should look like Figure 5.42.
3. Scale the helmet to fit the template. You will also need to move it back just a little, as shown in Figure 5.43.
4. The helmet has a metal lip around the bottom edge. Scale and move the vertices so the bottom row of faces forms the lip, as shown in Figure 5.44.
5. Select all the faces of the helmet except the bottom row and extrude faces. Scale the extrusion in as shown in Figure 5.45.

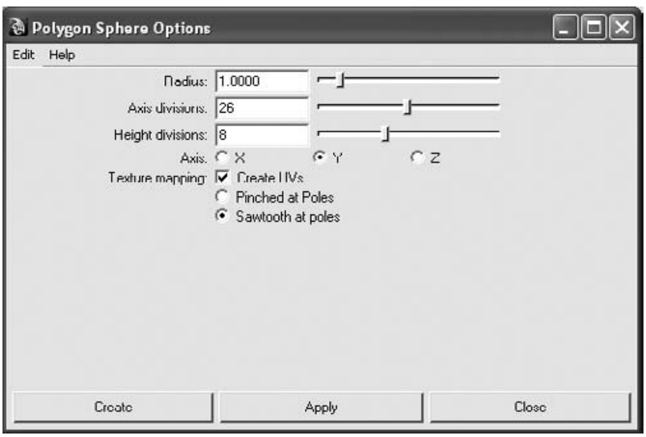


Figure 5.41
Create a polygon sphere for the helmet.

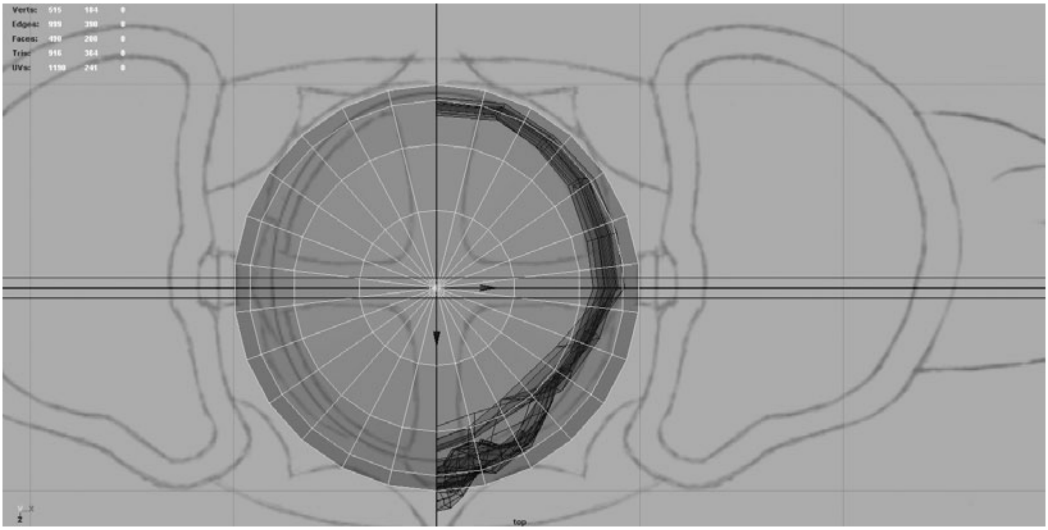


Figure 5.42
View the helmet from the top view for adjustments.

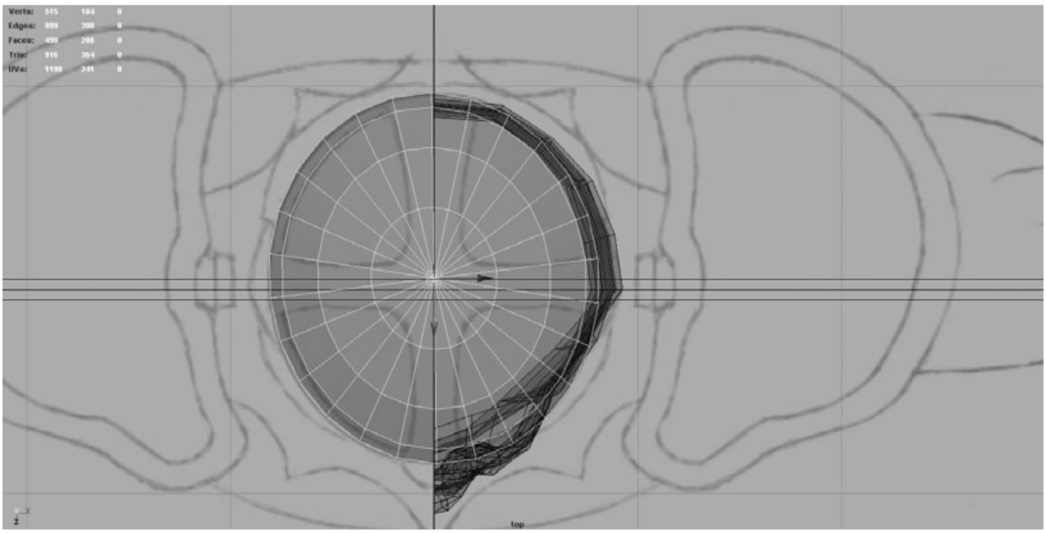


Figure 5.43
Move the helmet back just a little to better fit the template.

6. Notice the metal ridge that runs up the helmet from the base to the cap in the front, back, and sides. You will extrude faces to create these ridges, but before you can do that, you need to create the flares where the ridges meet at that op of the helmet and the bottom edge of the helmet. Use

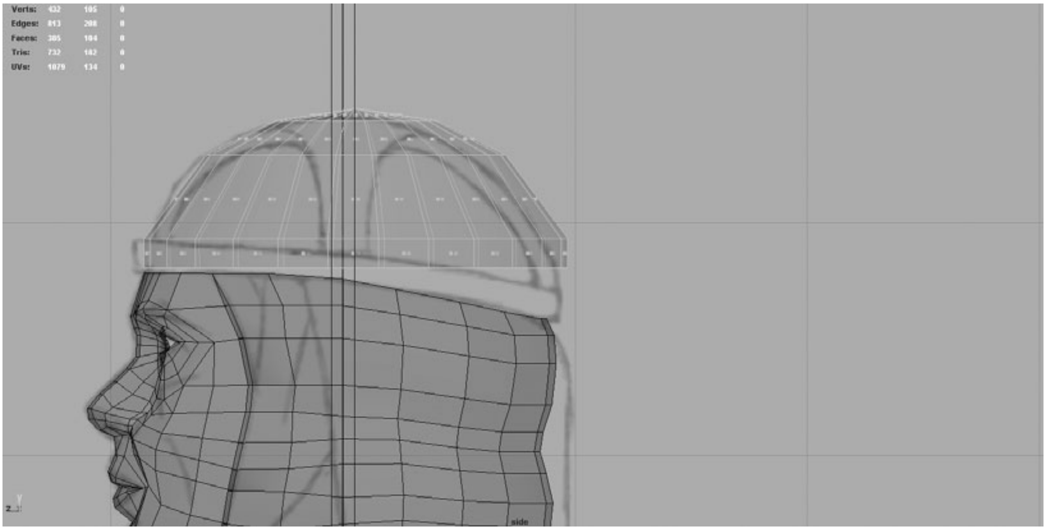


Figure 5.44
Form the bottom lip of the helmet.

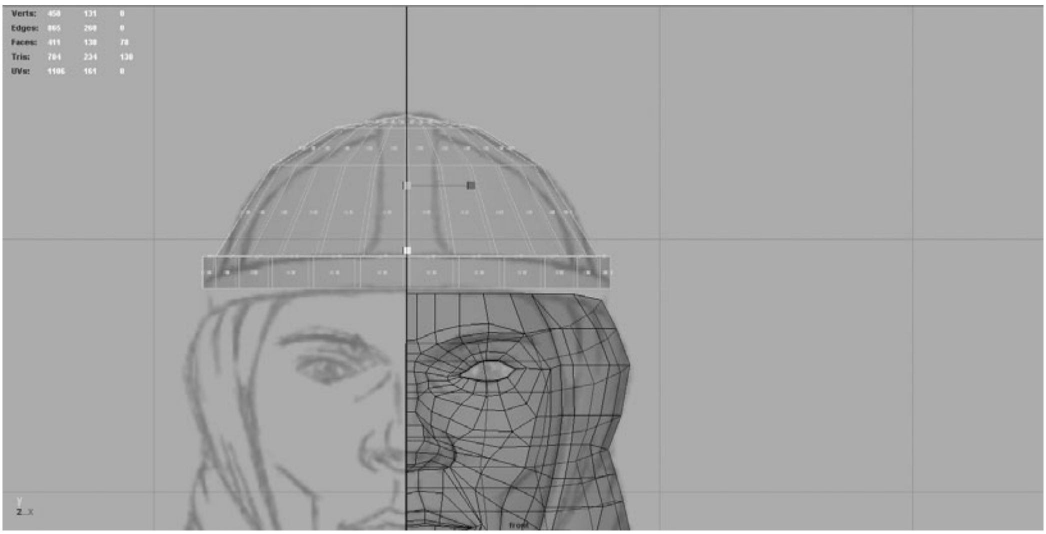


Figure 5.45
Extrude the upper faces of the helmet in.

the Split Polygon tool in the Edit Mesh menu to split the polygons as shown in Figure 5.46.

7. Split the polygons at the back of the helmet the same way you did those in the front.

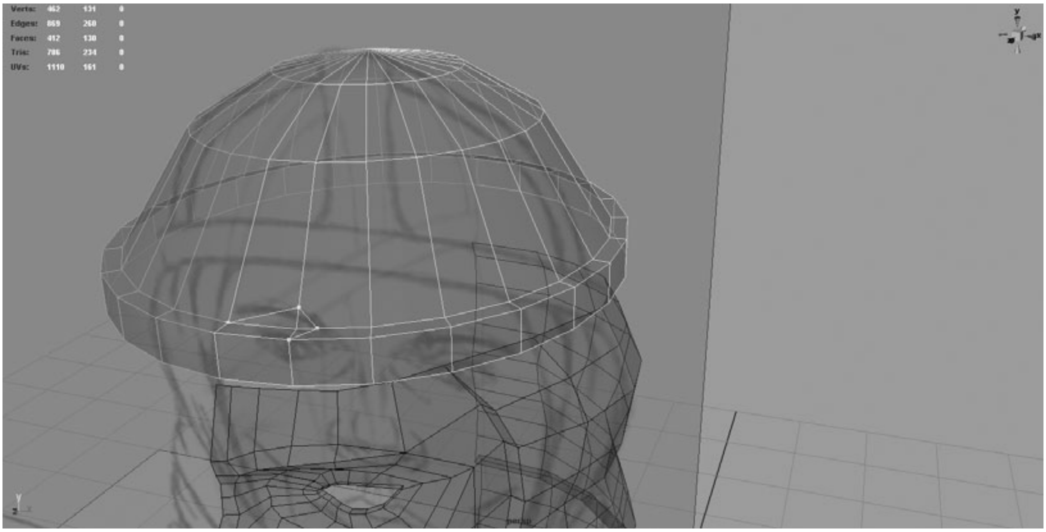


Figure 5.46
Split the polygons where needed to form the flares at the bottom of the helmet ridges.

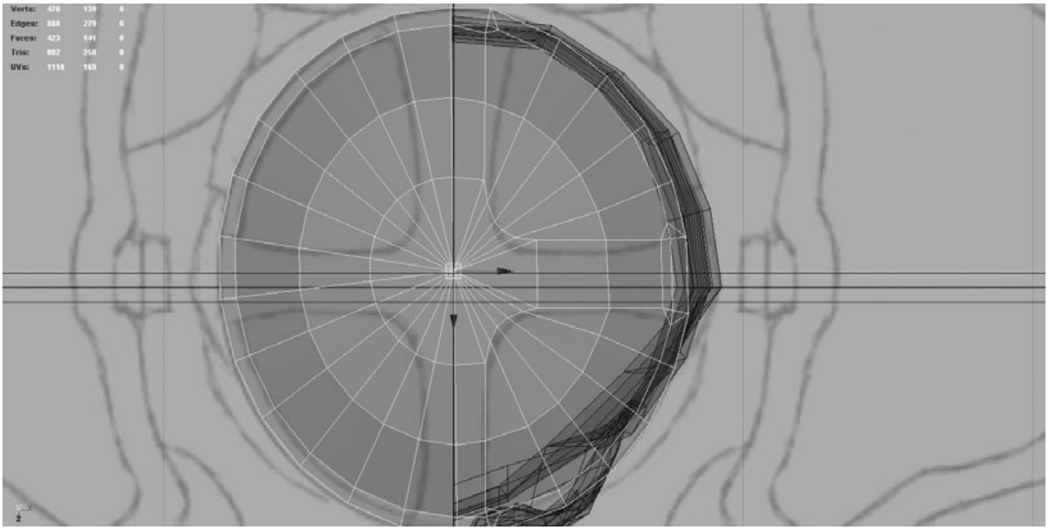


Figure 5.47
Move the vertices of the helmet to define where the ridges will be.

8. Split the polygons on the side of the helmet to form the flares at the base of the side ridge.

9. From the top view, move the vertices so the ridges are straight. Also adjust the vertices where the ridges meet at the top of the helmet. Use Figure 5.47 as a guide for defining the ridges and the area where they meet at the top of the helmet.

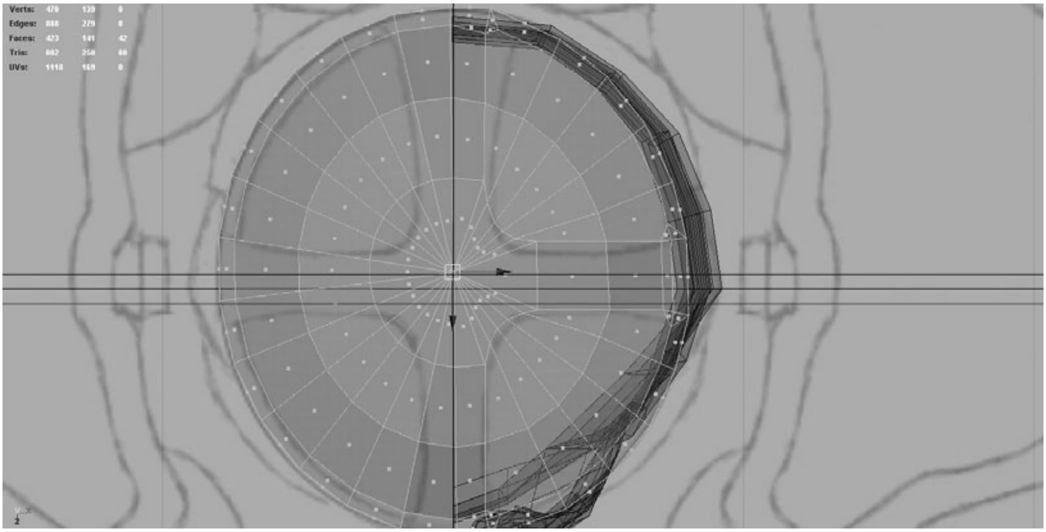


Figure 5.48
Extrude the helmet inward to form the ridges.

10. Select the faces that do not form the lip or ridges of the helmet and extrude them in as shown in Figure 5.48.

Note

You only need to worry about the faces on the right side of the screen because those on the left side will be deleted later. The faces on the left side of the screen are selected during the extrusion process so that the scale operation will work from the center of the helmet.

11. Switch to Perspective view and adjust the scaling of the helmet as shown in Figure 5.49. Try to make it as even as you can. You might need to scale a little more in the x axis than in the z axis to get it to look just right.
12. Delete the unneeded half of the helmet, as shown in Figure 5.50.
13. Move the vertices of the helmet to follow the template from the side view as shown in Figure 5.51.
14. Now the helmet and the head need to be joined into a continuous mesh. Select the edges along the bottom of the helmet and extrude them in to meet those of the head.
15. The vertices of the helmet and those of the head do not line up exactly, so you will have to adjust the ones in the helmet to match those of the head. Some vertices along the back of the helmet will need to be snapped together

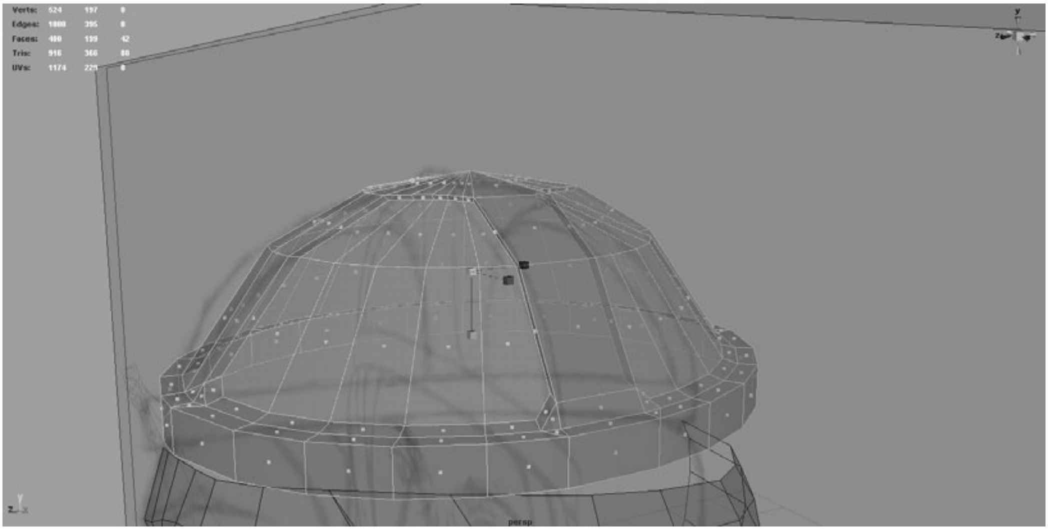


Figure 5.49
Adjust the scaling in the Perspective view.

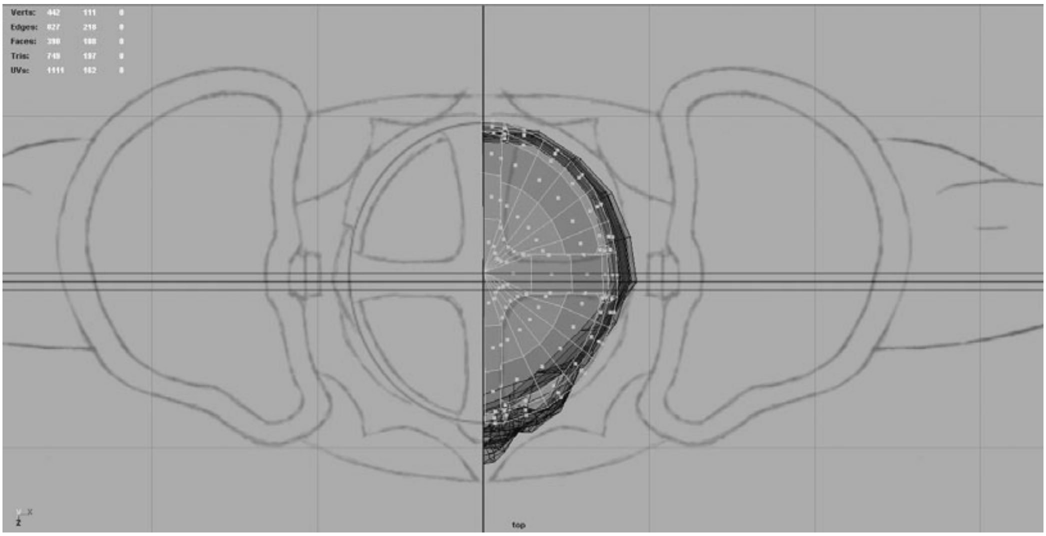


Figure 5.50
Delete the unneeded polygons of the helmet.

and some along the front will need to be split. You need to snap a vertex from the helmet to every vertex on the head to get them to join properly. See Figure 5.52 for an example.

16. Combine the helmet and head objects and merge the joining vertices to make a single mesh object as shown in Figure 5.53.

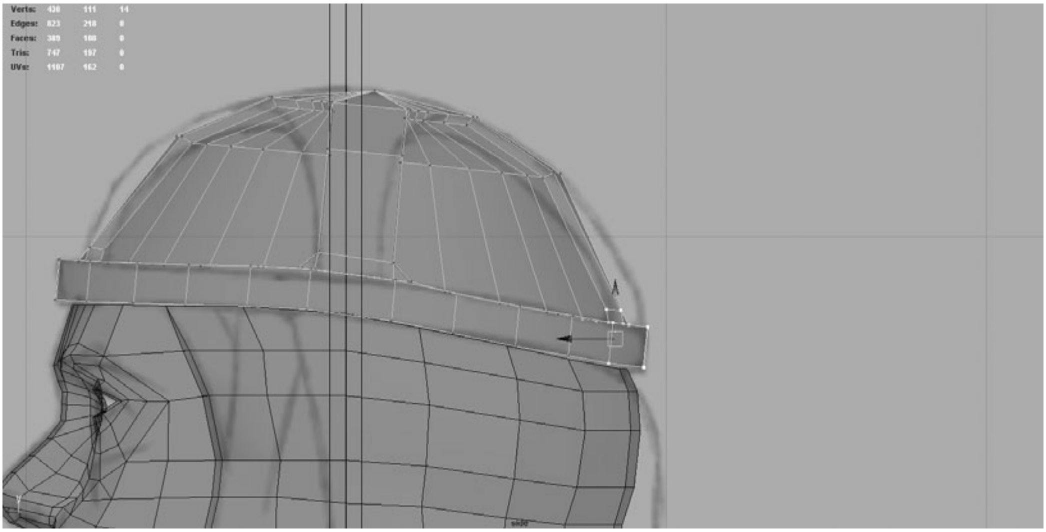


Figure 5.51
Move the vertices of the helmet to follow the template.

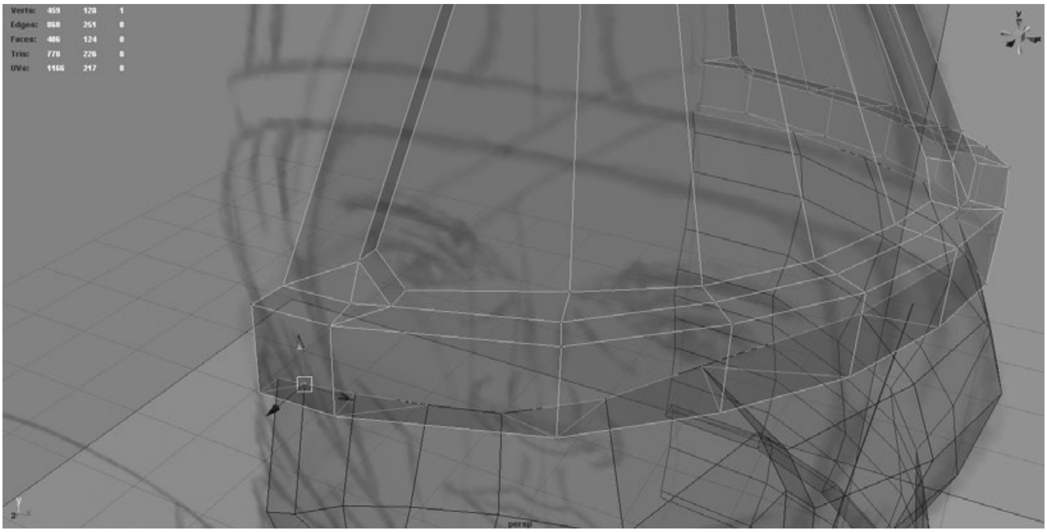


Figure 5.52
Snap the vertices of the helmet to those of the head.

17. The only thing left to model for the head is the chainmail on the back. Change the selection mode to Faces and select the row of faces just below the helmet as shown in Figure 5.54.

18. Extrude the selected faces and pull them out from the head just a little.

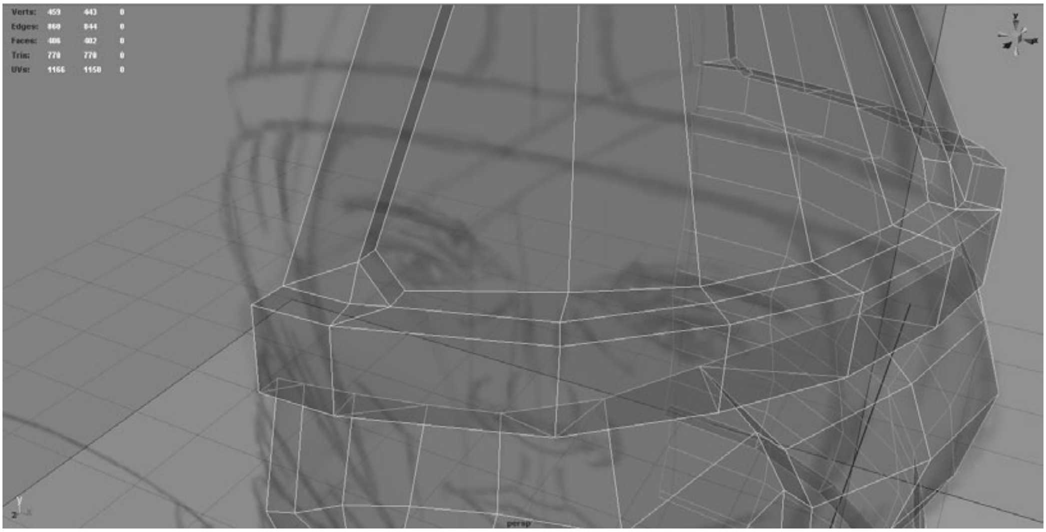


Figure 5.53
Make the helmet and head objects a single mesh.

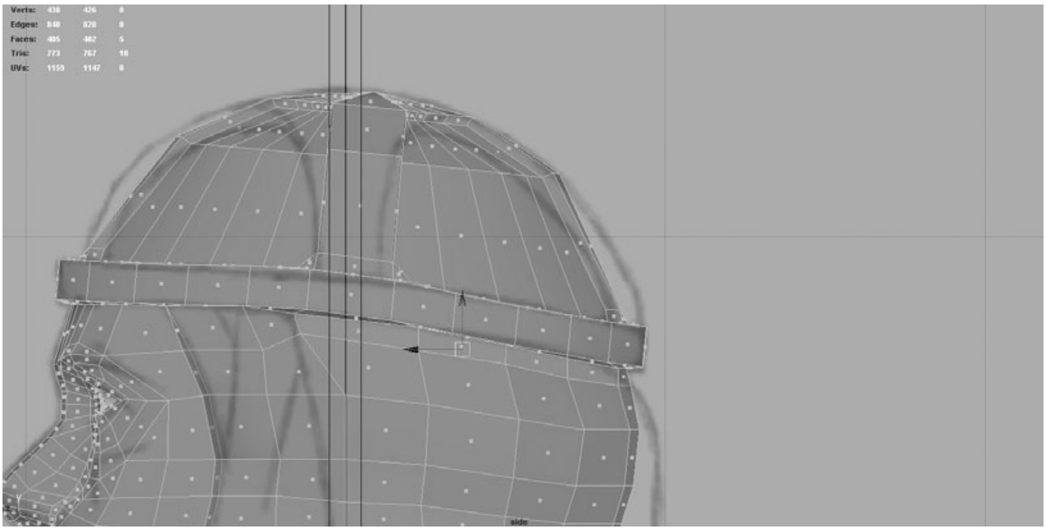


Figure 5.54
Select the faces just below the helmet.

19. Snap the bottom vertices of the extruded faces back to the vertices from which they originated and merge them with those vertices. The idea is to form a lip from which to extrude the chainmail, as shown in Figure 5.55.

20. Select the faces of the lip and extrude down to create the chainmail. Have the chainmail extend out around the wrap as shown in Figure 5.56.

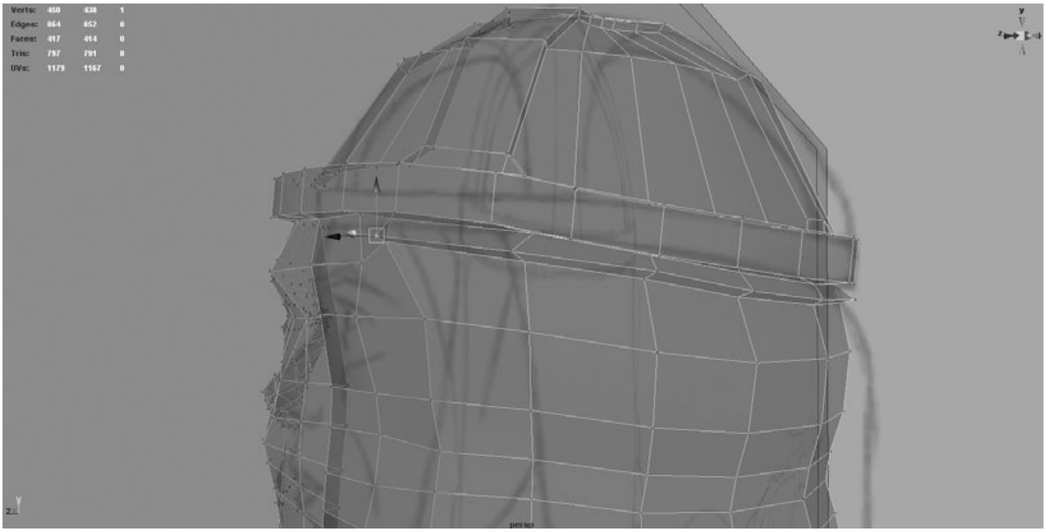


Figure 5.55
Create a lip from which you can extrude the chainmail.

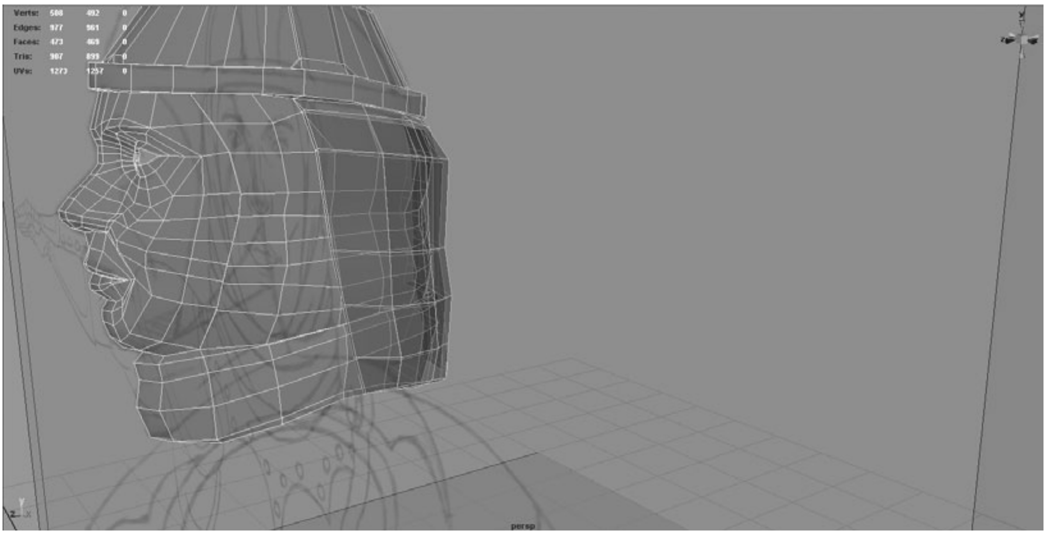


Figure 5.56
Extrude the faces of the lip to create the chainmail.

Note

> The chainmail could have been modeled as part of the wrap, but that would have limited the animation and made it look less realistic. With the wrap and chainmail separated, the chainmail can be animated with the motion of the character.

Although you are almost finished with the character's head, there are still a few things you need to model such as the inside of his mouth and his eyeballs; you'll

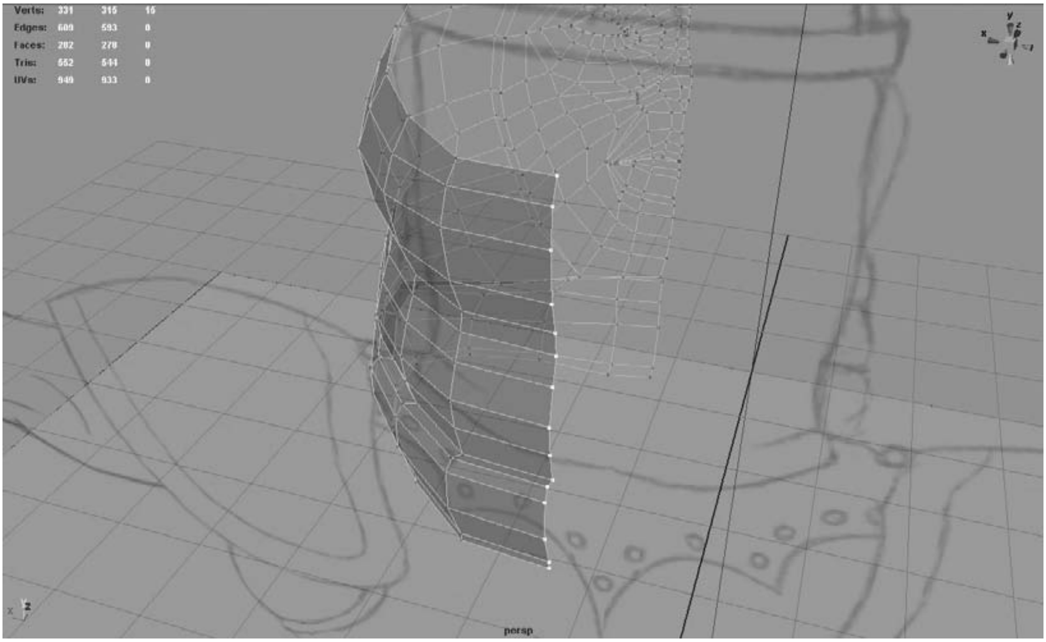


Figure 5.57
Snap the vertices along the center line to line up directly over the line.

come back to those later. For now, to finish getting the head ready, you need to make sure that the vertices along the center line of the head are lined up directly over that line. While holding the X key down, move the vertices along the center line in just the x axis to snap them to the center line, as shown in Figure 5.57.

Modeling the Breast Plate

The breast plate of this character is made of metal and is a rigid object. It will not bend and twist like other parts of the character. In addition to remembering this when rigging and animating the character, it is also important to recognize it during the modeling process.

1. The breast plate of your character tucks up under the neck wrap. Start modeling it by selecting the edges at the bottom of the wrap as shown in Figure 5.58.

2. From the side view, extrude the edges down the course of the breast plate to where it goes under the character's belt. With each extrusion, use the Scale tool to adjust the breast plate to fit the template, as shown in Figure 5.59. Try to match your model with the example here.

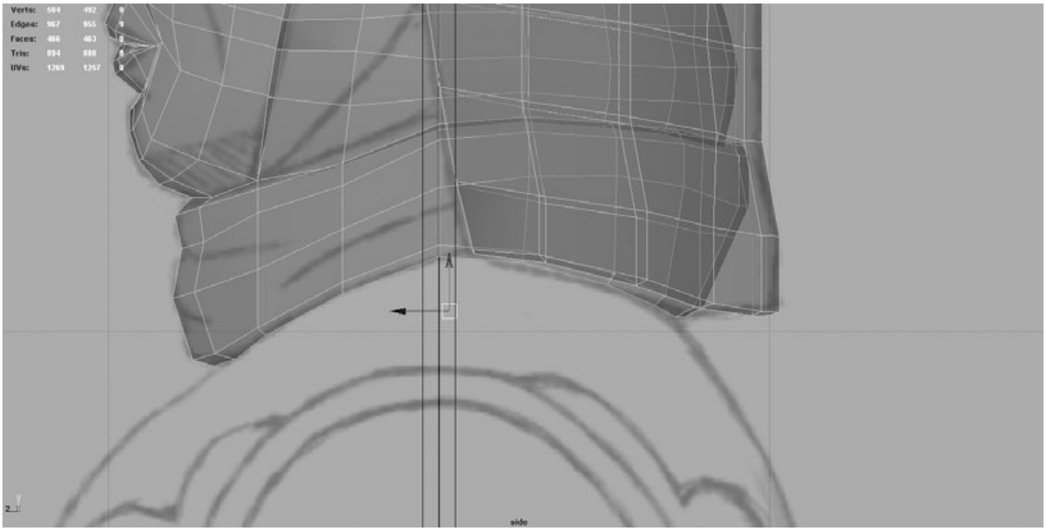


Figure 5.58
Select the bottom edges of the head model.

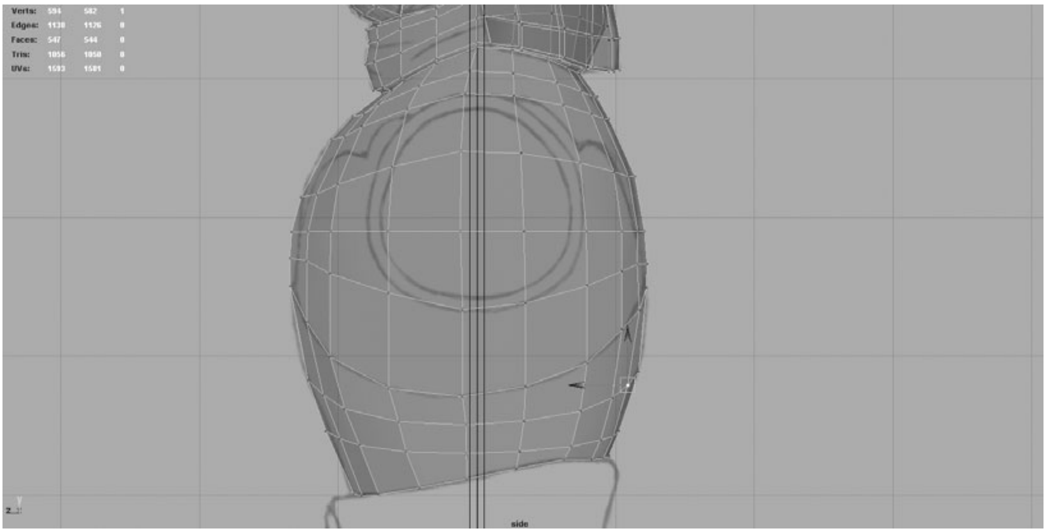


Figure 5.59
Extrude the edges to create the polygons of the breast plate.

3. Go to the front view and move the vertices to follow the lines of the breast plate as shown in Figure 5.60.

4. Have the vertices along the opening for the arm follow the edge of the breast plate as shown in Figure 5.61.

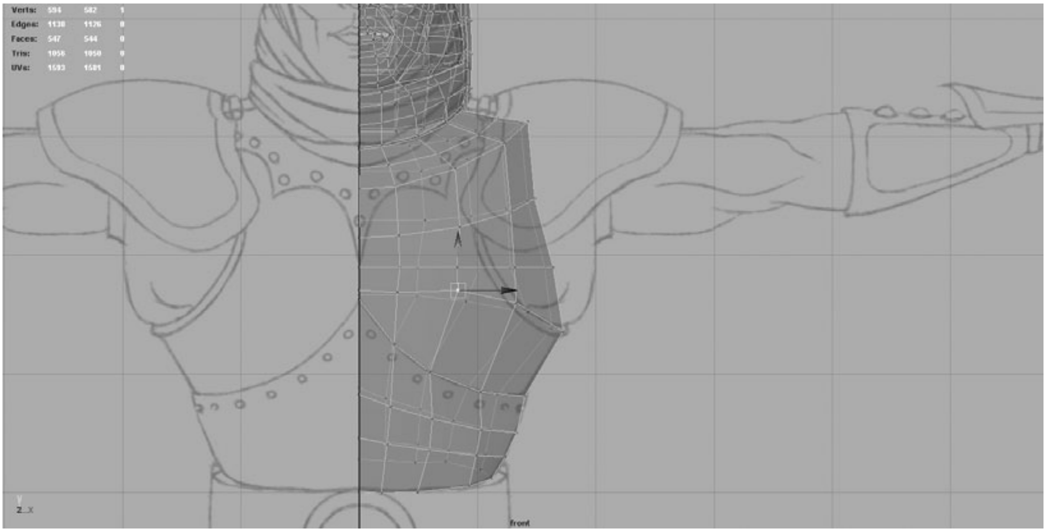


Figure 5.60
Adjust the vertices to of the breast plate to fit the template from the front.

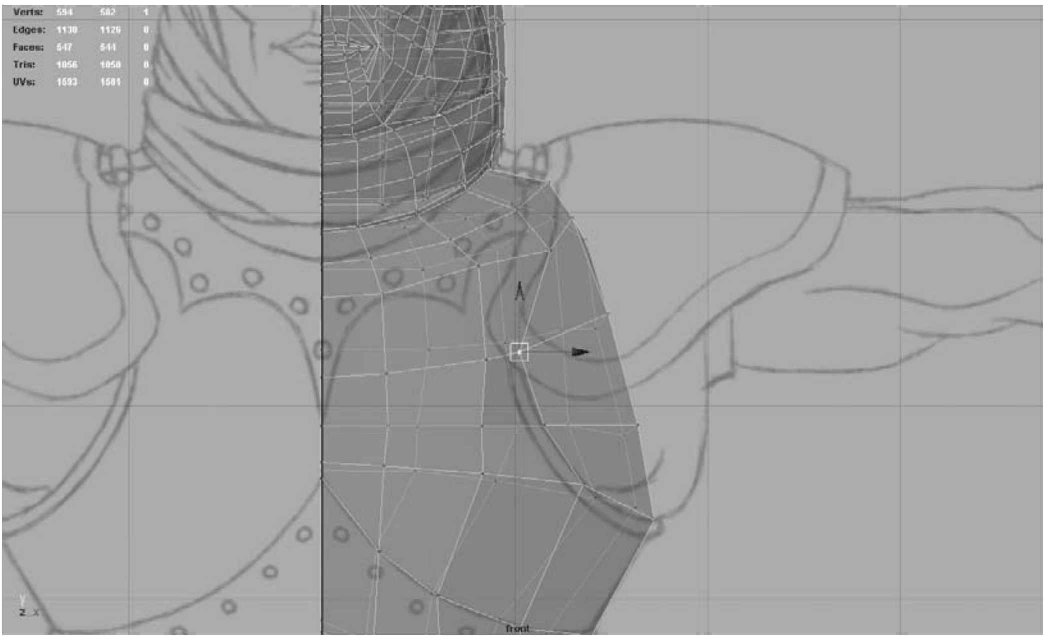


Figure 5.61
Have the vertices follow the edge of the breast plate.

5. You need to adjust the vertices on the back of the character, but as you may have noticed, there is no back view. To create one, Open the Panels menu, choose Orthographic, select New, and choose Front. This creates a new camera in the scene. In the Channel box, rename the new camera Back1. To move the camera around to the back, set the Y rotation to 180 and set Translate Z to −100.1, as shown in Figure 5.62

6. Finish adjusting the vertices of the breast plate to match those of the arm opening from both the back and side views, as shown in Figure 5.63.

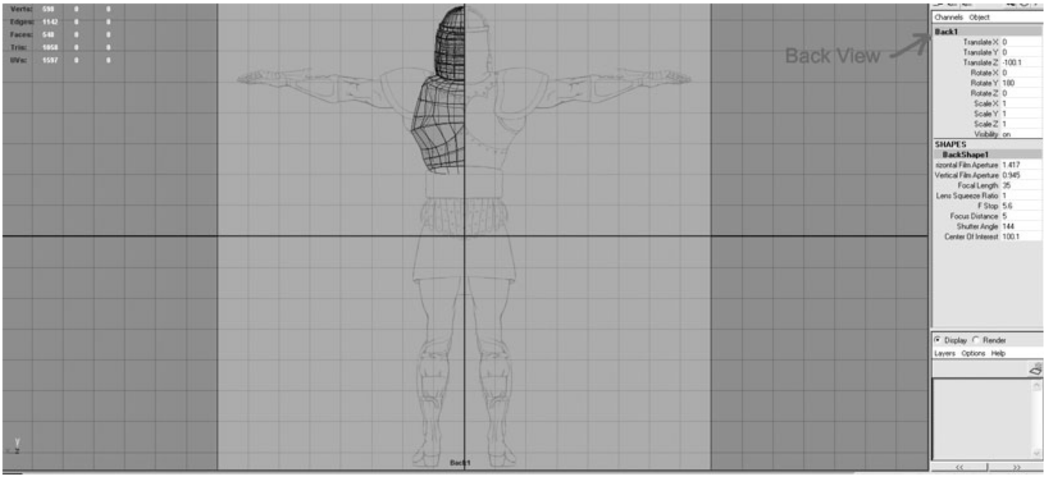


Figure 5.62
Create a back-view camera.

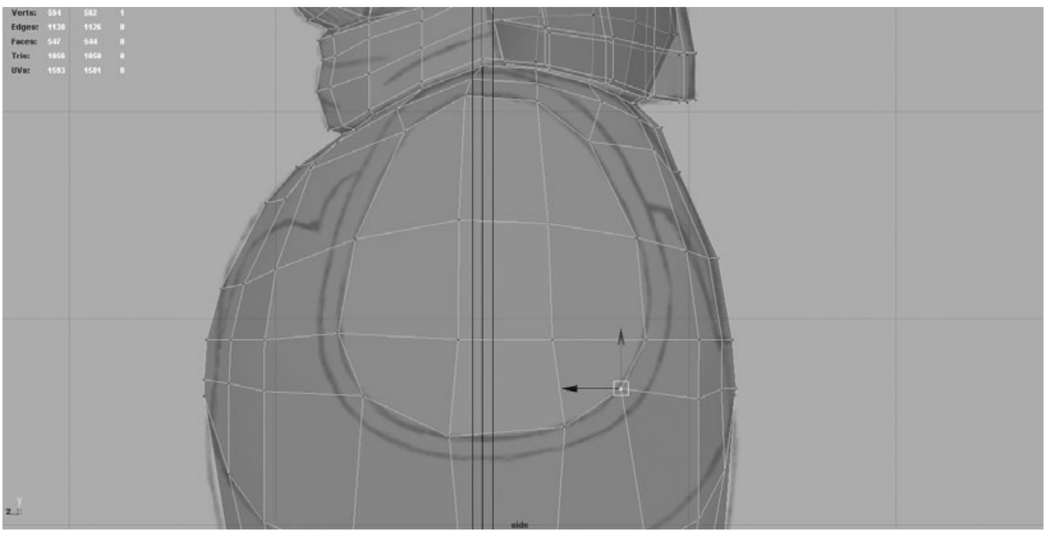


Figure 5.63
Move the vertices around the arm opening in the breast plate.

That's pretty much it for the breast plate. Check it over in the Perspective view to make sure you have the shape right. Notice that the breast plate also has a shoulder plate attached to it with a hinge at the shoulder; because the plate fits over the arm and shoulder, you'll build that part of the model after you create the arm.

Modeling the Arm

You'll model the arm by extruding the faces on the inside of the breast plate's arm opening.

1. In the side view, select the faces that make up the area inside the arm opening of the breast plate.
2. Switch to front view and begin to extrude the faces following the template, as shown in Figure 5.64. As you extrude the faces, use the Scale tool to flatten the extrusion.
3. Continue extruding the arm from the front view, following the template as shown, using the Scale and Move tools to adjust each extrusion (see Figure 5.65).
4. Go to the top view and, in Control Vertex selection mode, move the vertices of the arm to fit the template in that view as shown in Figure 5.66.

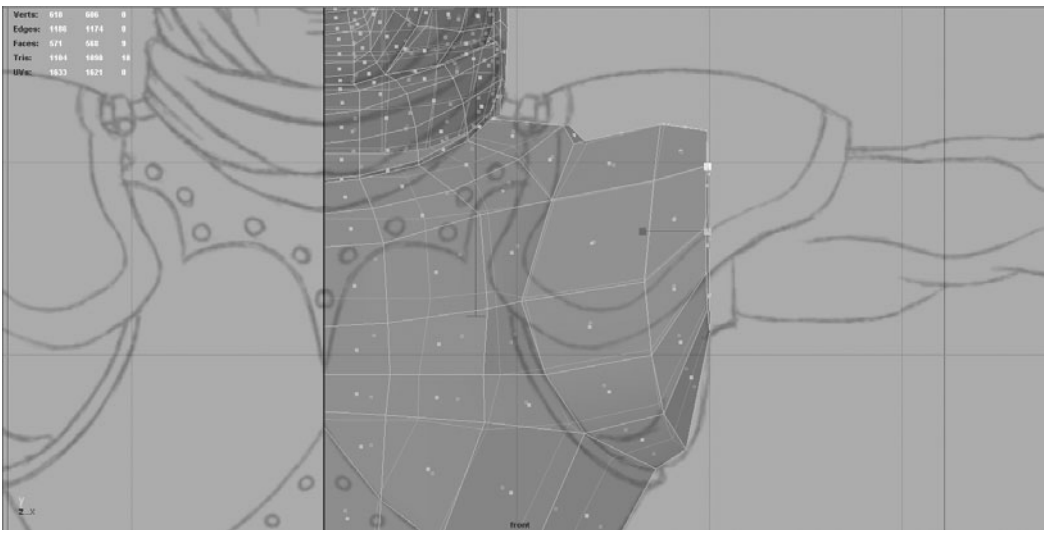


Figure 5.64
Start extruding faces for the arm.

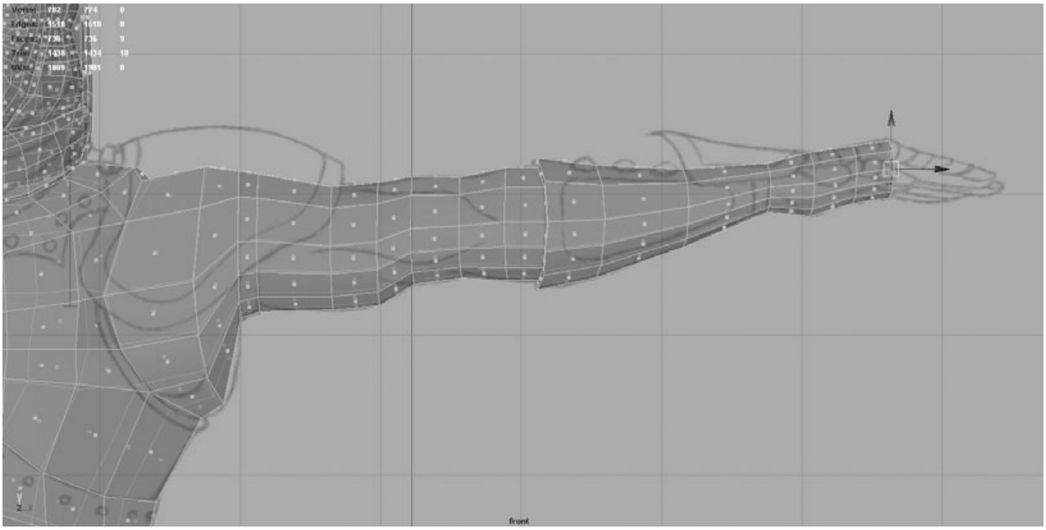


Figure 5.65
Extrude faces along the length of the arm.

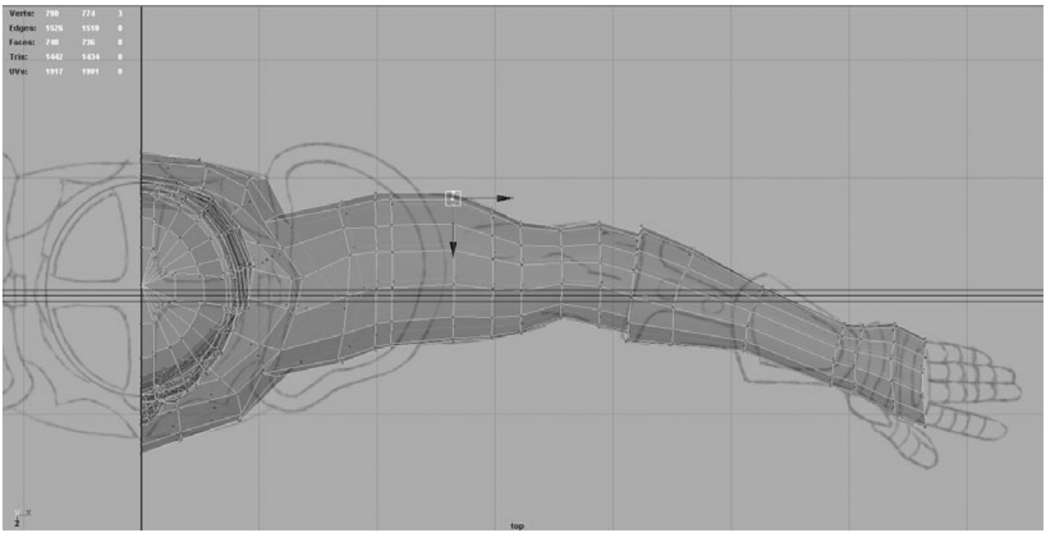


Figure 5.66
Move the vertices of the arm in the top view.

5. To build the shoulder plate, first create a separate polygon pipe object as shown in Figure 5.67. Change the Radius setting to 1, the Height setting to 4, and the Thickness setting to 0.1. Also, change the Subdivisions Around Axis setting to 12, the Subdivisions Along Height setting to 4, and the Subdivisions on Cap setting to 1. Create the object in the x axis with textures turned on.

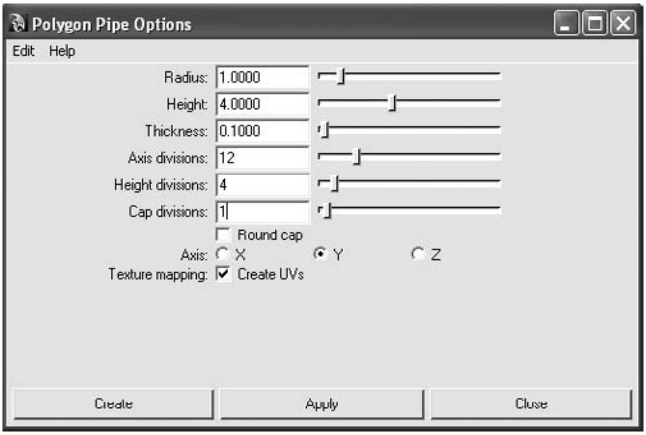


Figure 5.67
Create a polygon pipe object for the shoulder plate.

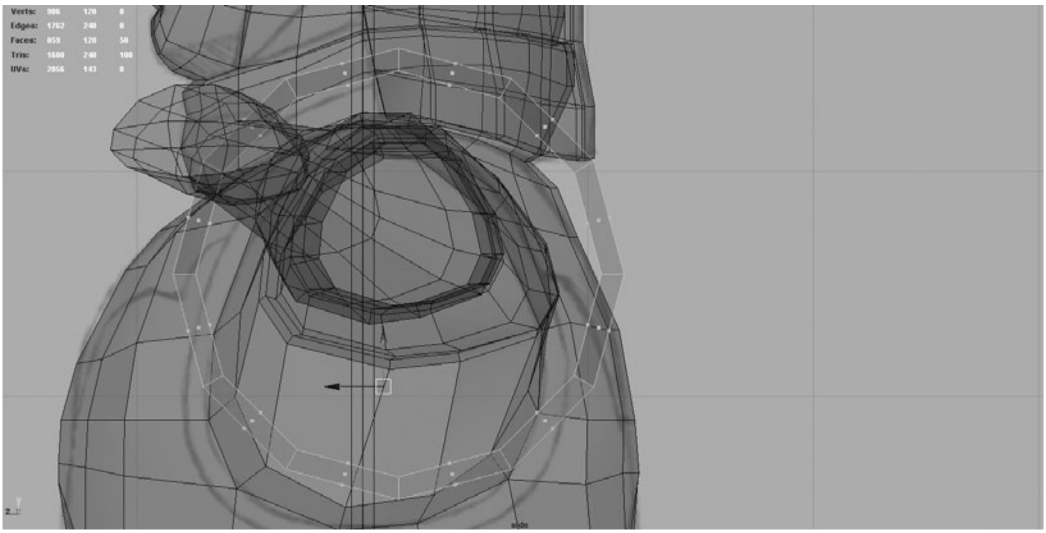


Figure 5.68
Delete the bottom faces of the pipe object.

6. Move the pipe object into position over the shoulder.
7. From the side view, select the faces shown in Figure 5.68 and delete them.
8. In the top view, move the vertices of the pipe object to follow the shape of the shoulder plate as shown in Figure 5.69.

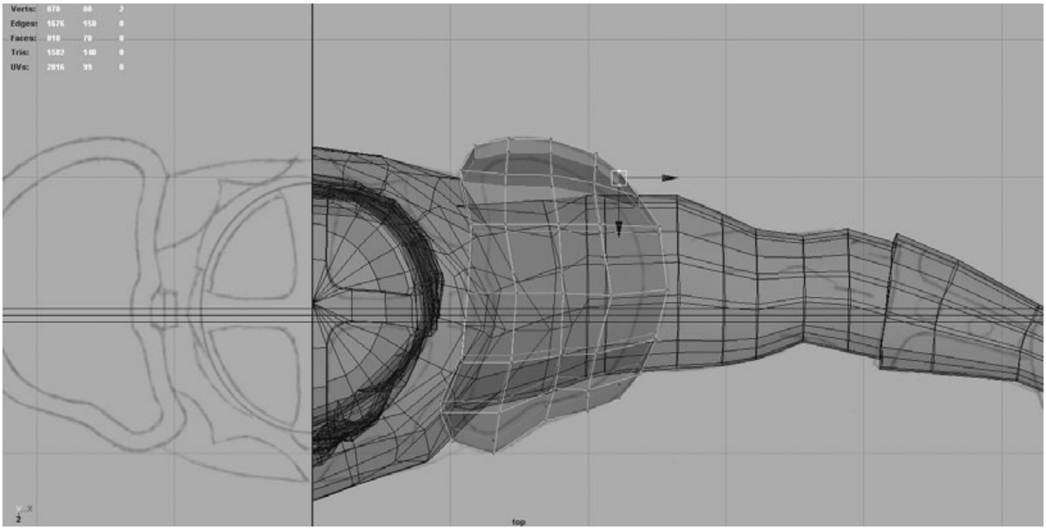


Figure 5.69
Move the vertices to form the shape of the shoulder plate.

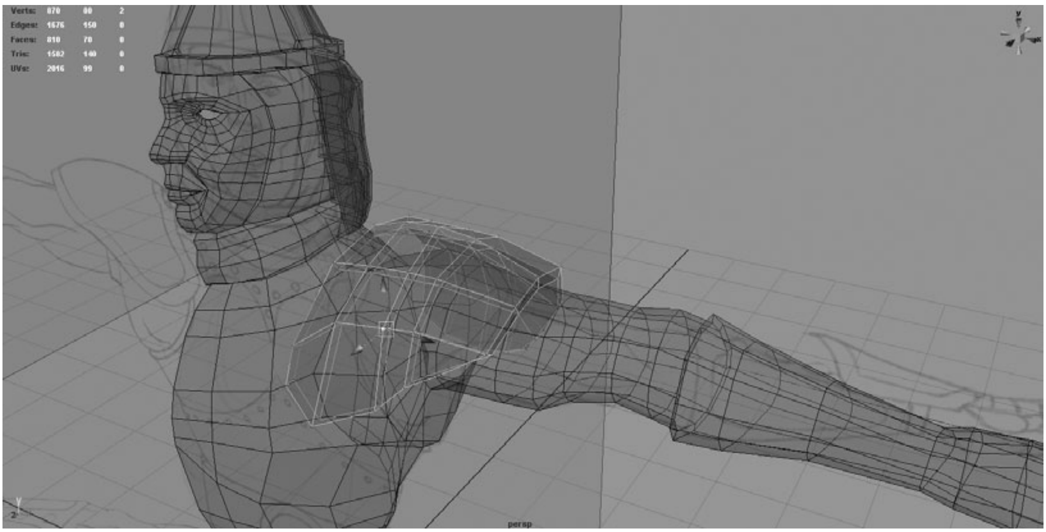


Figure 5.70
Move vertices to get the shoulder plate to the correct shape.

9. Adjust the vertices in the front, back, and Perspective views, making sure that the shape looks correct and that it does not intrude on the model of the breast plate or arm (see Figure 5.70).

10. Snap the top vertices along the edge of the shoulder plate to the bottom vertices to create a sharp edge, as shown in Figure 5.71.

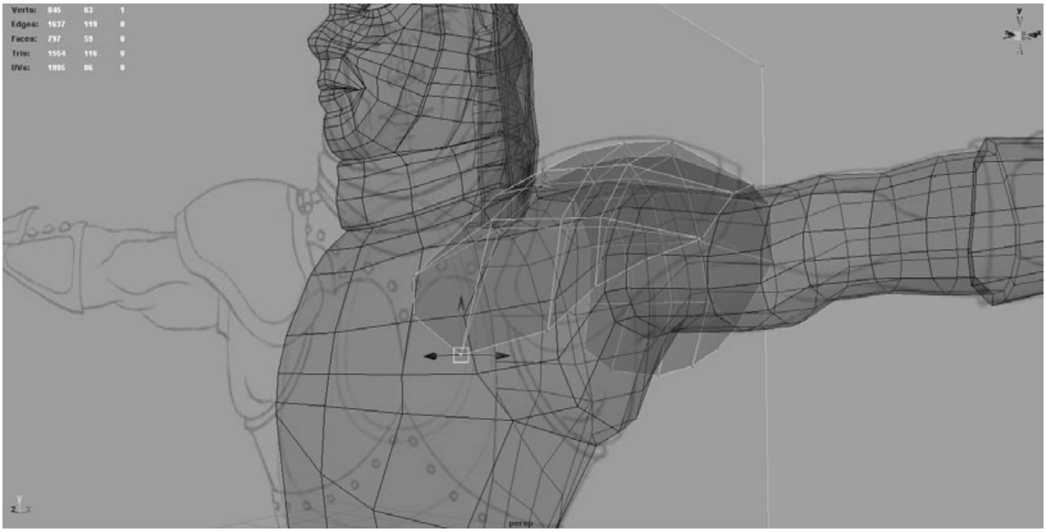


Figure 5.71
Snap the vertices of the edges together.

11. Merge the edge vertices.

That pretty much takes care of it for the arms. Double-check the model in Perspective view to fix any anatomical problems and to clean up any mistakes that might have occurred in the modeling process.

Tip

It is a good idea to look at your model from many angles as you model. You can often find mistakes that are easy to fix now but will become burdensome later.

Modeling the Hand

The character's hand is covered by gantlets, which have many metal segments. Rather than modeling each segment, however, let the textures carry that part of the model and just worry about the main geometric features of the hand.

1. Start with the wrist plate that extends out over the forearm. Move the vertices of the forearm to create a lip from which to extrude the wrist plate, as shown in Figure 5.72.
2. Extrude the faces from the lip and then adjust them to follow the shape of the wrist plate, as shown in Figure 5.73.

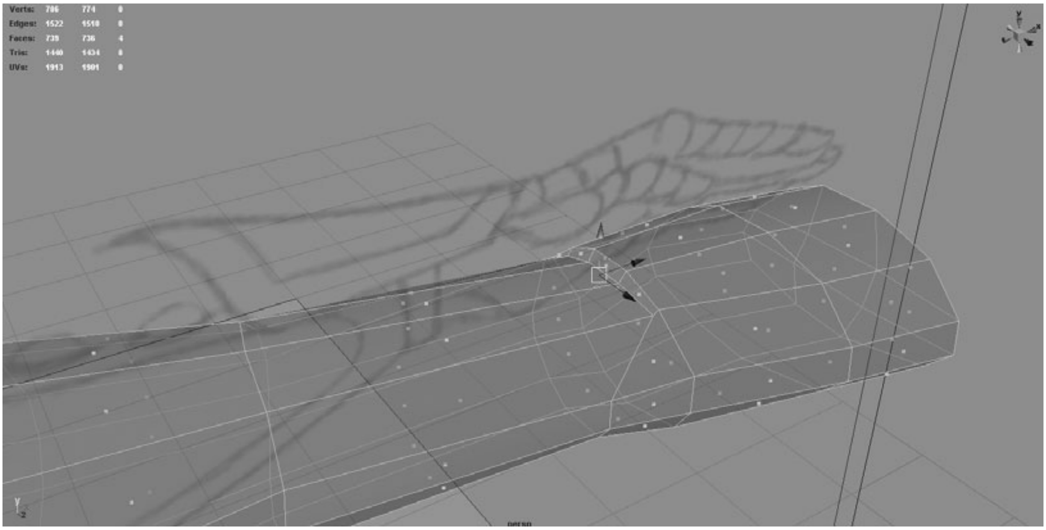


Figure 5.72
Create a lip for the wrist plate.

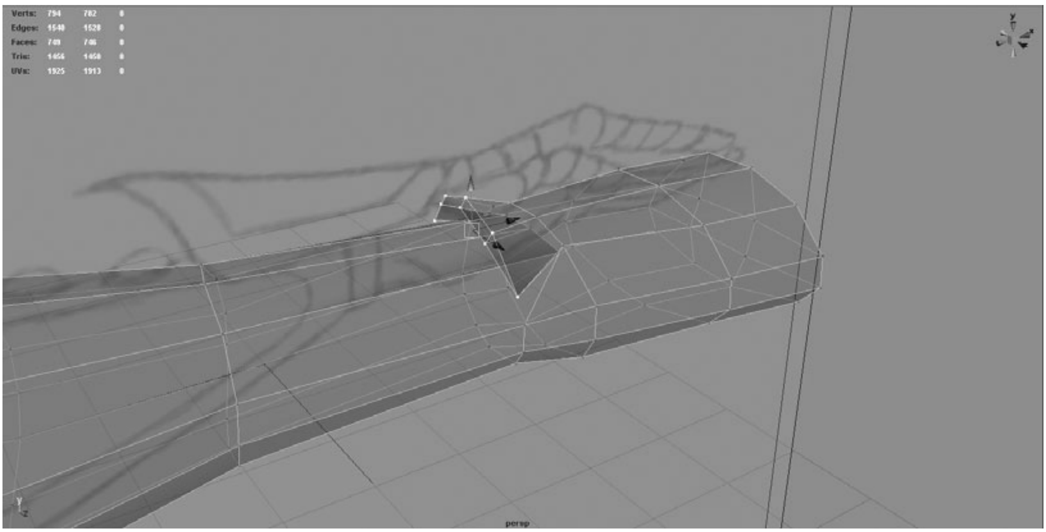


Figure 5.73
Start extruding the wrist plate.

3. Extrude and shape the wrist plate following the template in the front and top views (see Figure 5.74).
4. Move the vertices of the palm to create areas for extruding the thumb and fingers. The palm also needs to look more like the palm of a hand. Although

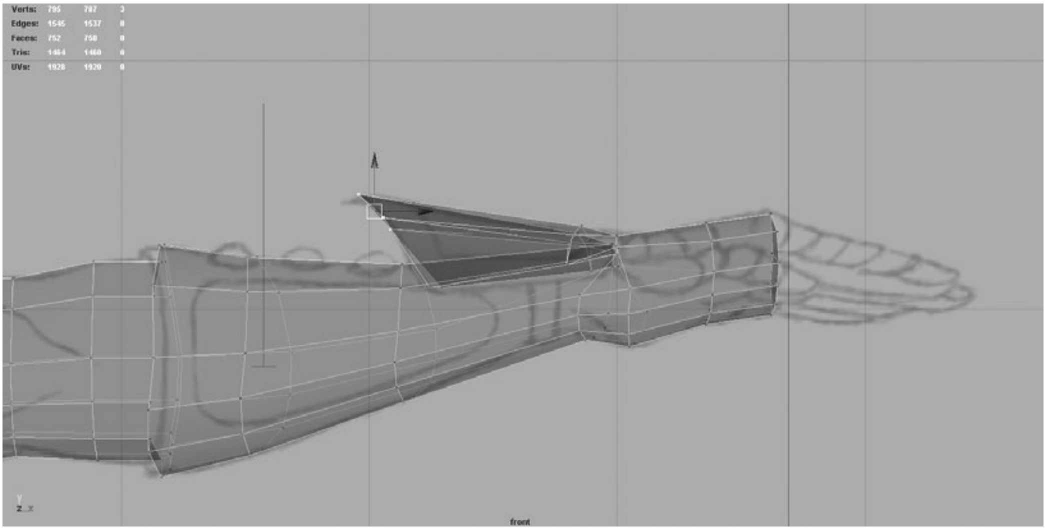


Figure 5.74
Create the wrist plate.

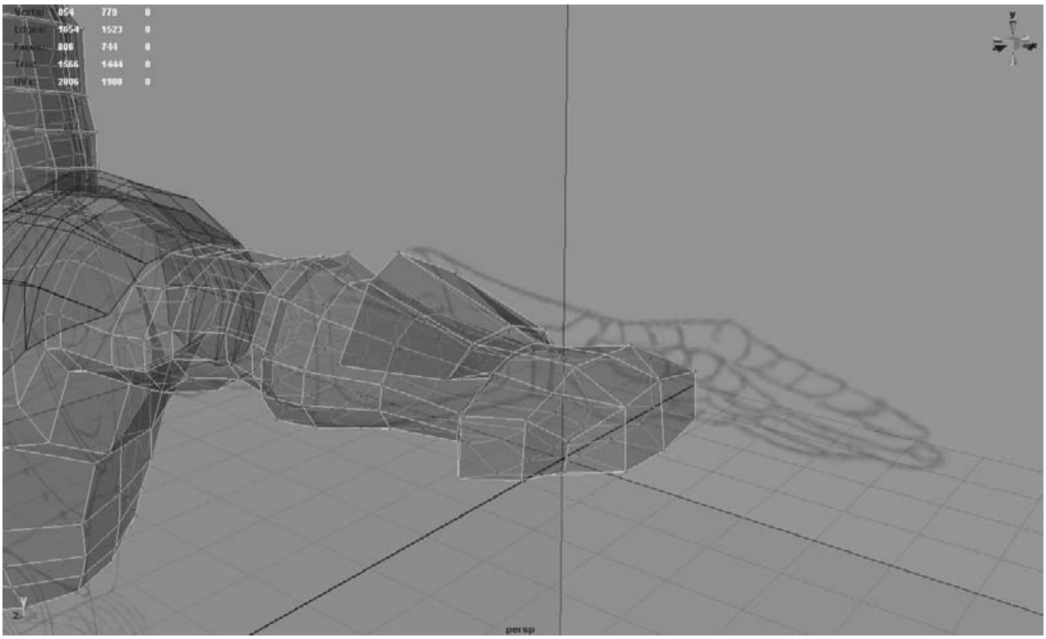


Figure 5.75
Move vertices in the palm to gain the correct shape.

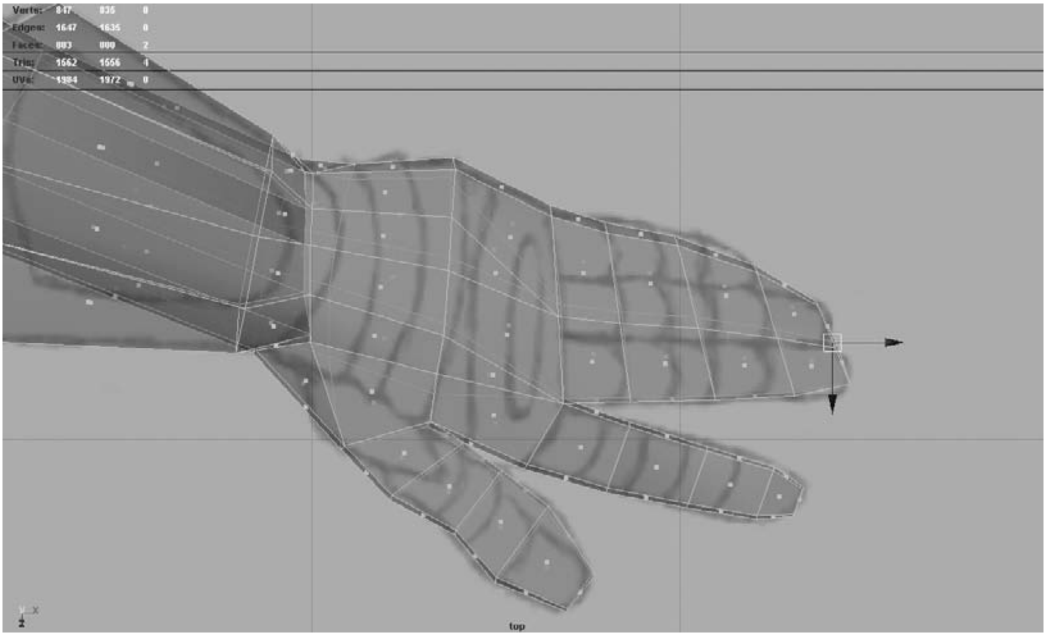


Figure 5.76
Extrude the thumb and fingers.

there is no way to show this in a screen shot, Figure 5.75 does show the palm after I adjusted the vertices.

5. From the top view, extrude faces for the thumb, index finger, and three other fingers. Use Figure 5.76 as a guide for the number of extrusions.

Note

Some characters may need to have full-hand articulation and thus require separate modeling for each finger. Your warrior needs only limited articulation, so modeling the three fingers as a single unit will work fine and will prevent you from having to create a lot of extra polygons in the model.

6. Scale and move the vertices of the hand to look correct anatomically, as shown in Figure 5.77.

You might notice that the thumb and fingers are very blocky, but to model them completely would take a lot of extra polygons. Because these polygons are very small, you can get away with simplifying the hands. If the character's hands take a more prominent role in the game, however, you might need to make them more complex.

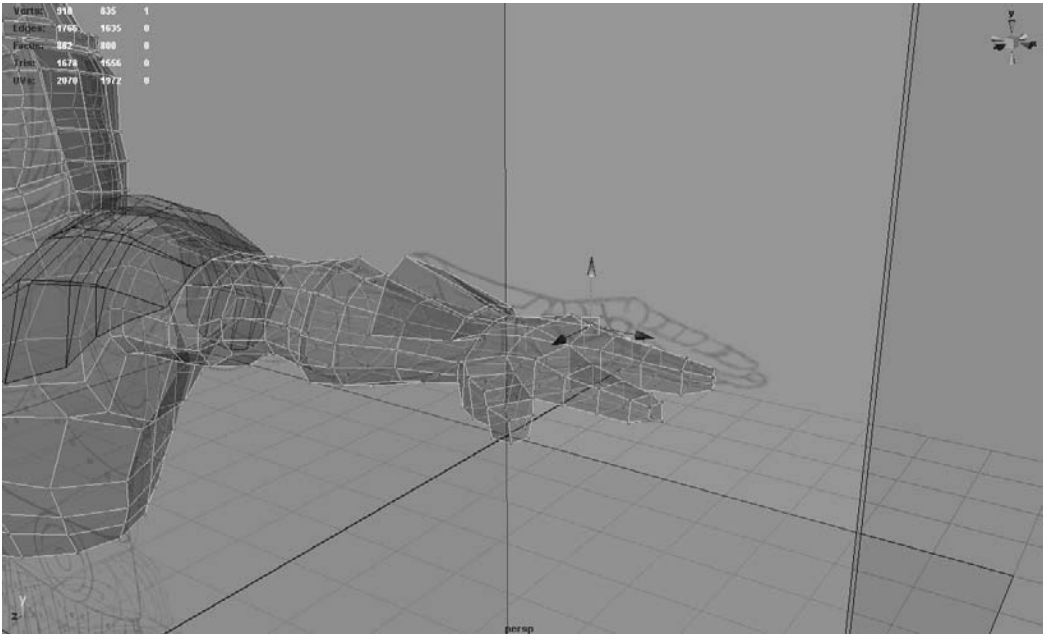


Figure 5.77
Move the vertices in other views to finish the hand.

Modeling the Leg

This section covers modeling both the hips and the legs because they are connected.

1. Select the edges at the bottom of the breast plate and extrude them to the bottom of the character's chainmail tunic, as shown in Figure 5.78. Adjust the vertices as shown in the figure.

2. Go to the side view and move the vertices to follow the template from that view (see Figure 5.79).

3. You need to indicate in the geometry the leather straps extending from the character's belt. To do so, move the vertices so the edges of the polygons line up with the angle of the straps as closely as possible. In instances where moving vertices is impractical, split the polygons as shown in Figure 5.80.

4. Extend the edge that follows the bottom of the straps around the back of the character, as shown in Figure 5.81.

5. Select the area of the model where the leather straps are and extrude those faces as shown in Figure 5.82.

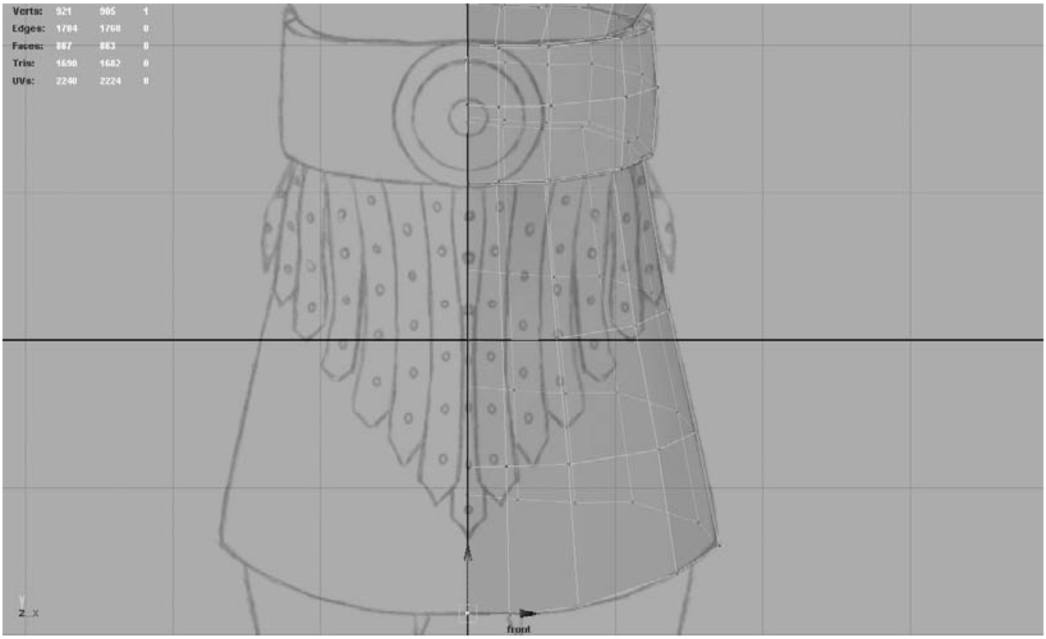


Figure 5.78
Extrude the edges of the model to the bottom of the tunic.

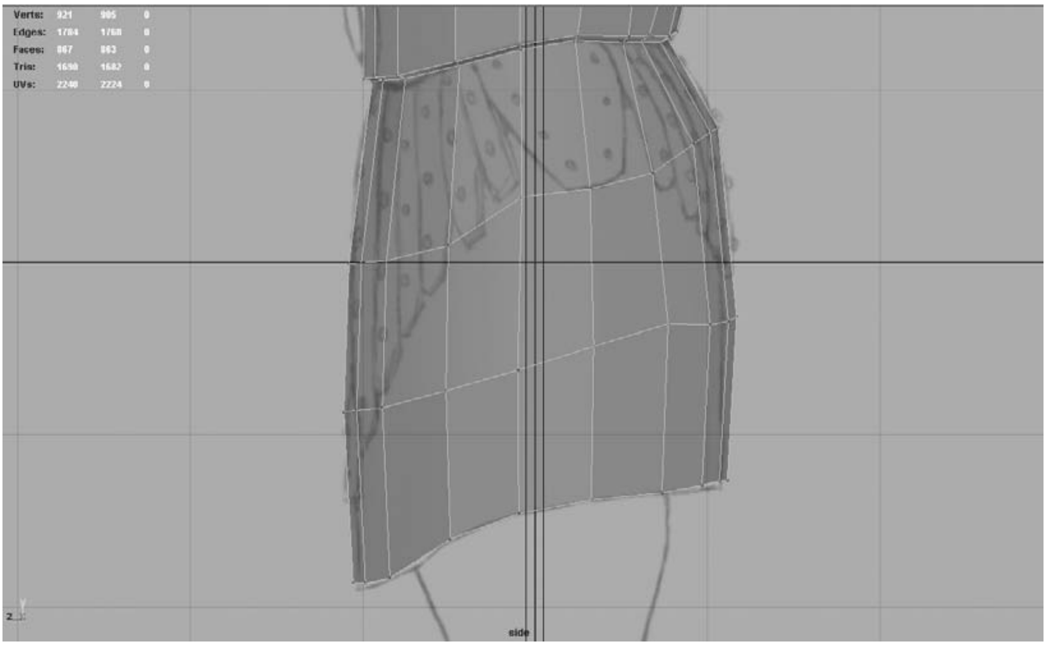


Figure 5.79
Move the vertices to follow the template in the side view.

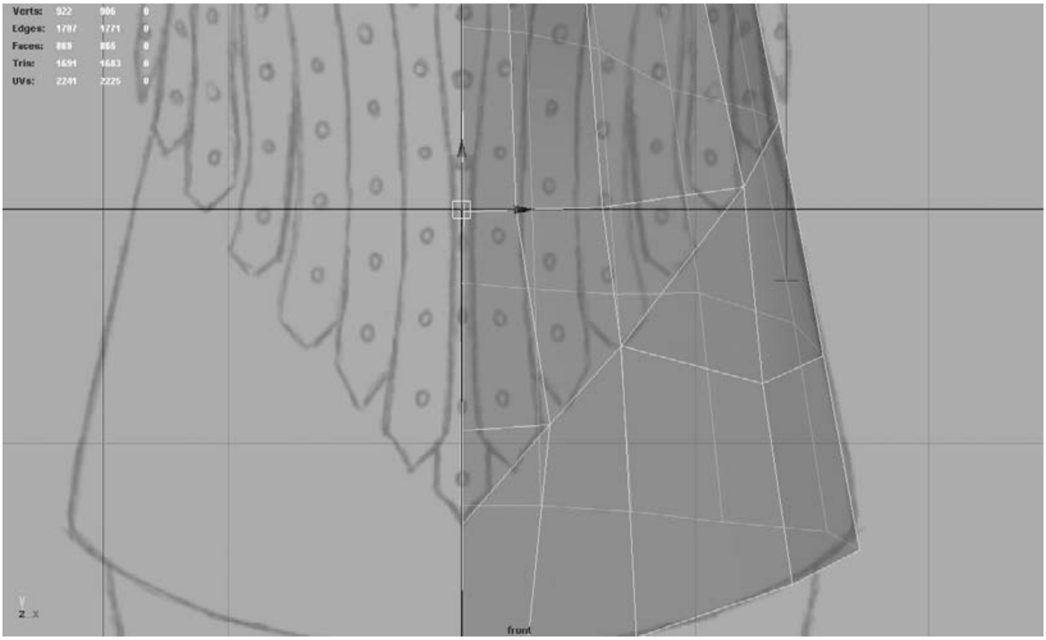


Figure 5.80
Split some of the polygons to create an edge along the bottom of the leather straps.

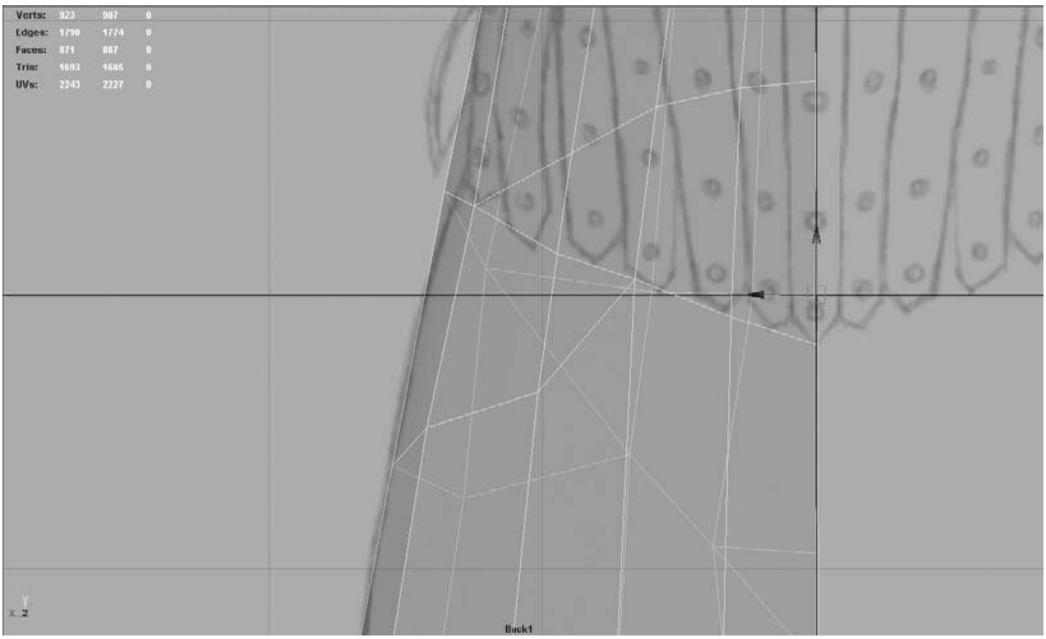


Figure 5.81
Continue the edge along the back as well.

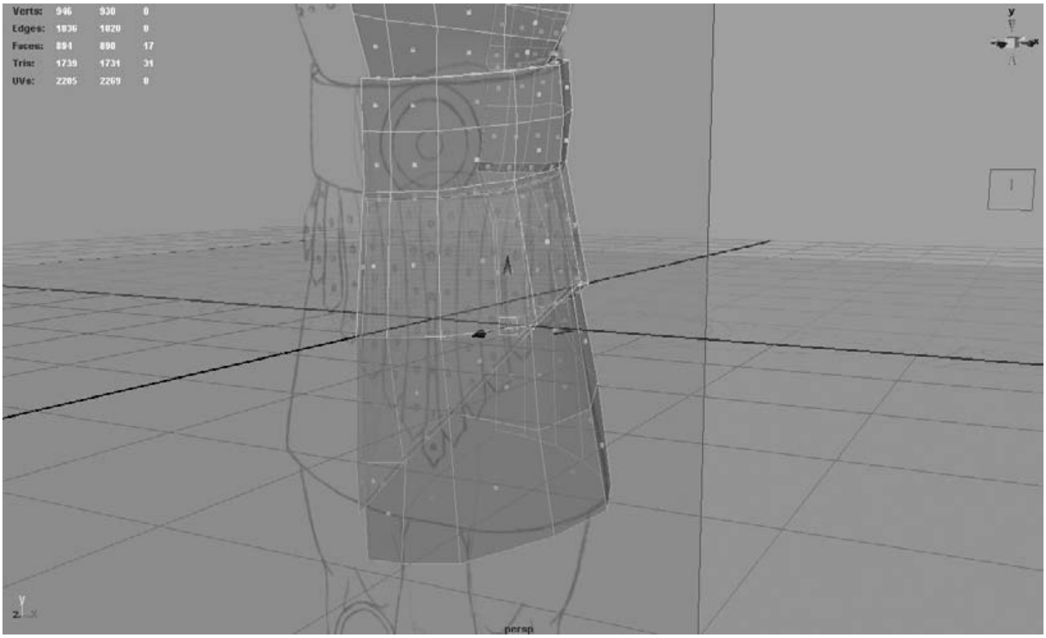


Figure 5.82
Extrude the faces of the leather straps.

6. Snap the vertices along the top of the extrusion back to the base of the belt and merge those vertices as shown in Figure 5.83.
7. Now you need to create some polygons extending from the bottom of the tunic to the character's legs. Select the bottom edges and extrude them up and in as shown in Figure 5.84. To avoid problems later when animating the character only go up the legs a little bit as shown.
8. Select the edges at the front and back of the model along the center line and extrude them toward each other, as shown in Figure 5.85.
9. Merge the vertices to join the back with the front, enclosing the top of the leg.
10. Although there is no template to guide you in shaping of the leg, you do need to move the vertices to form this area as best you can. You can use Figure 5.86 as an example.
11. Select the edges around the leg and make several extrusions down the length of the leg. Add a few extra at the knee so it will animate more accurately. Look at Figure 5.87 as a reference for the extrusions.

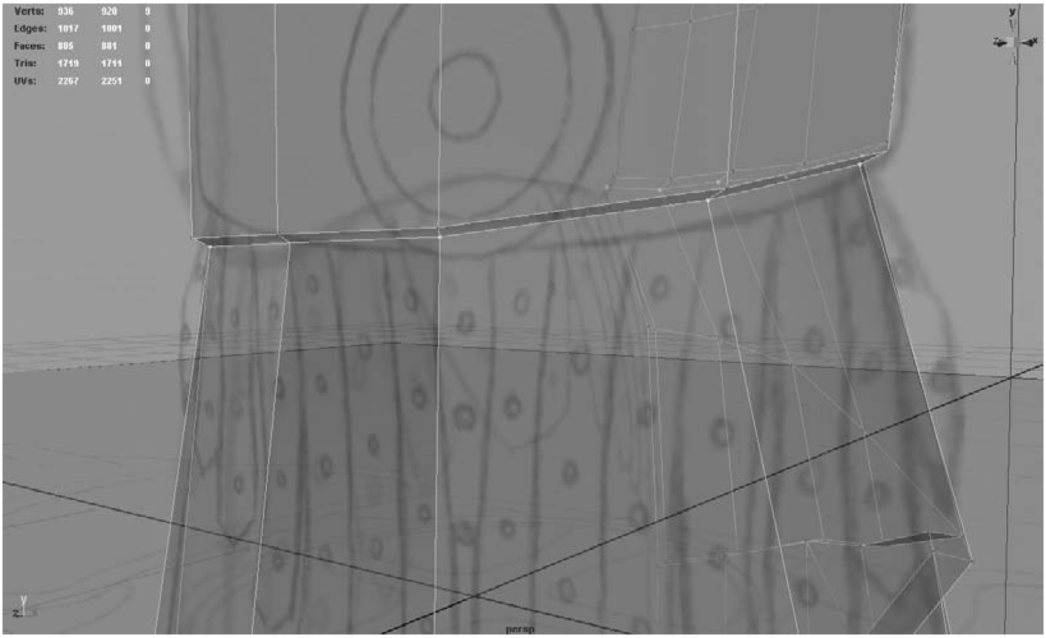


Figure 5.83
Merge the vertices of the top of the extrusion with those at the bottom of the belt.

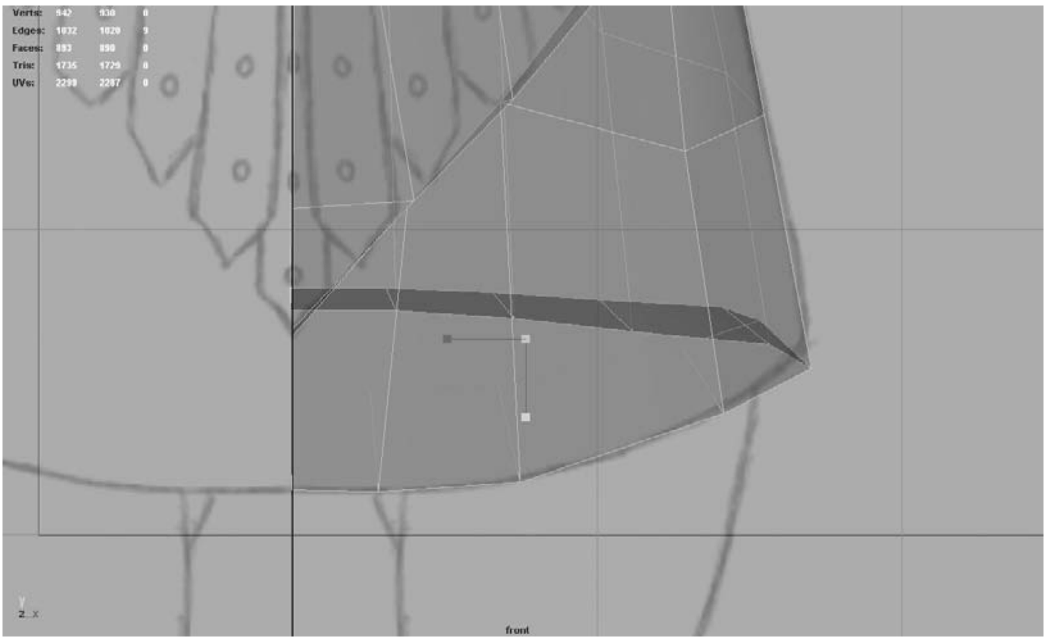


Figure 5.84
Extrude edges from the tunic to the character's leg.

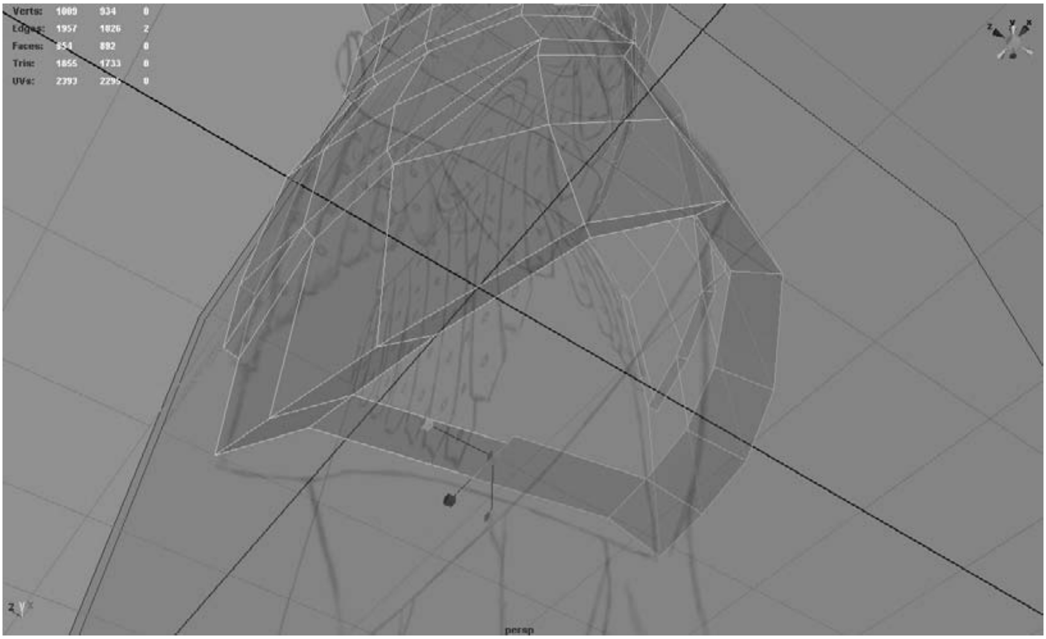


Figure 5.85
Extrude the edges along the center line to enclose the leg.

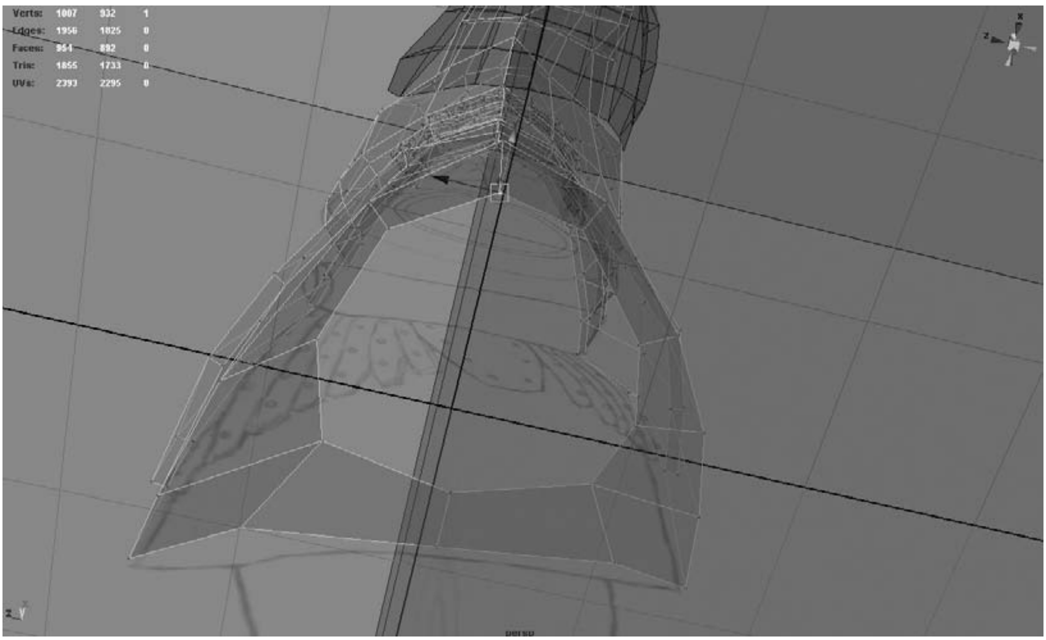


Figure 5.86
Move the vertices of the top of the leg to follow the shape of the leg.

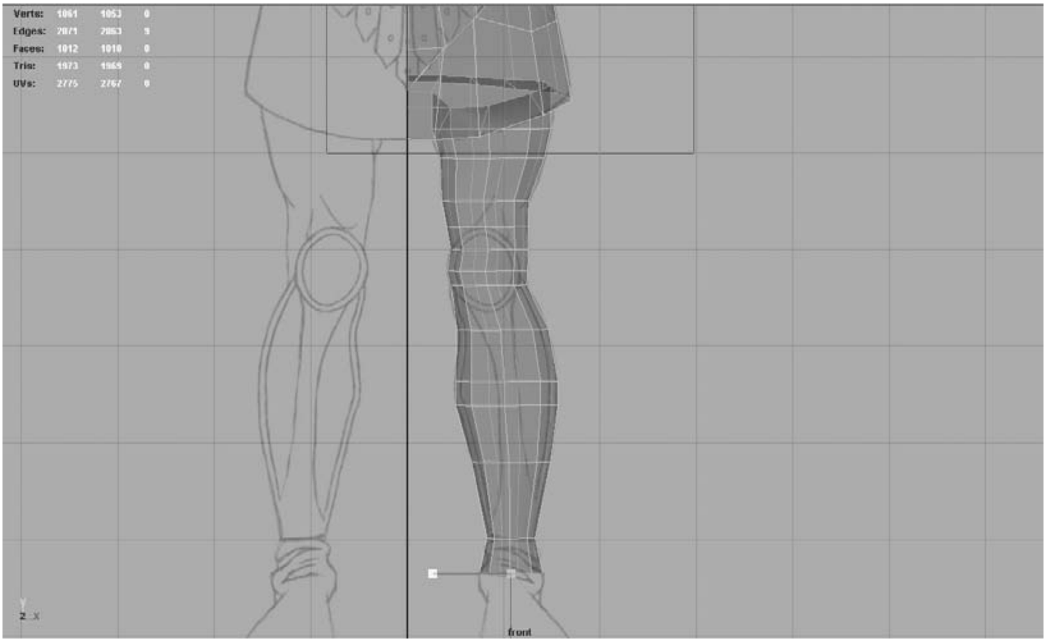


Figure 5.87
Extrude the edges of the leg following the template.

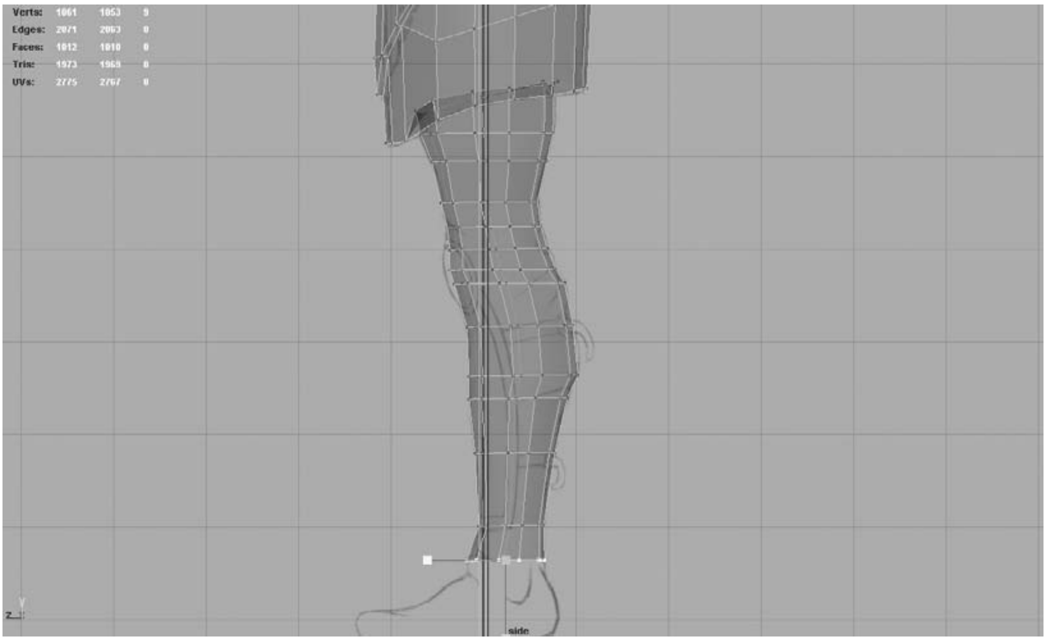


Figure 5.88
Adjust the vertices of the leg from the side view.

12. Like you did with the arm, go to a different view and adjust the vertices of the leg to fit that view as well (see Figure 5.88).

How do you like your model so far? Is it shaping up? You are getting close to being finished! You have just a few more things to model.

Modeling the Foot

Because the character is wearing boots, the foot is a lot easier to model than the hand was, but it is still complex because of the curve of the heel.

1. From the side view, extrude the bottom edges of the leg around the heel of the foot down to the toes. You will need to not only use the Scale and Move tools but also the Rotate tool as well (see Figure 5.89).
2. Join the vertices of the toe and merge them, as shown in Figure 5.90.
3. Move around the foot in the Perspective view, adjusting the shape of the foot as you go. You need to create an arch on the inside of the foot and taper the outside of the foot. The foot should also spread out near the toes, as shown in Figure 5.91.

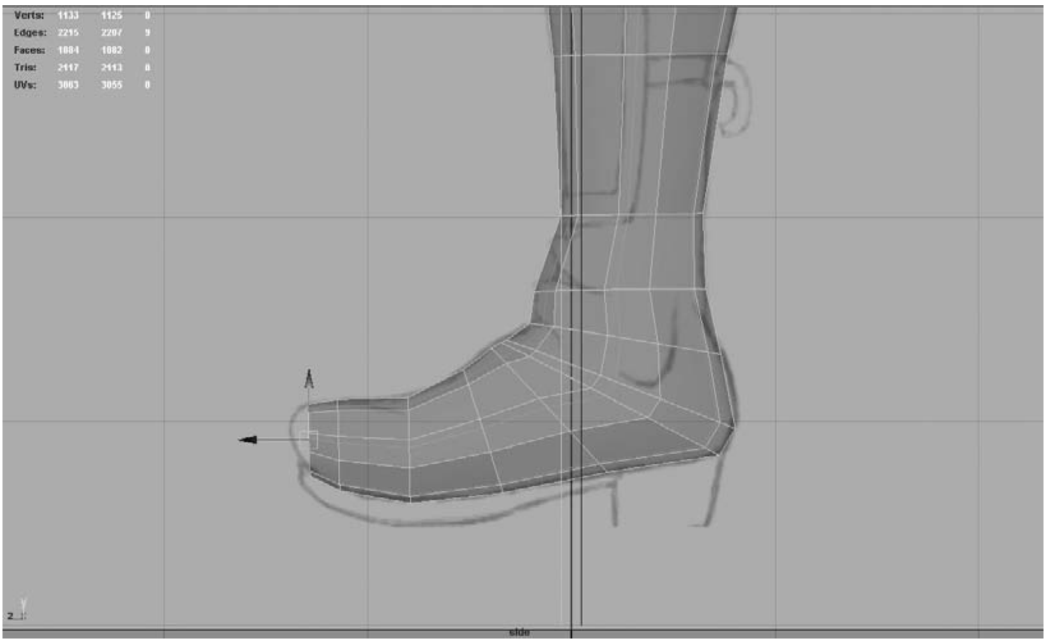


Figure 5.89
Extrude the edges to form the foot of the character.

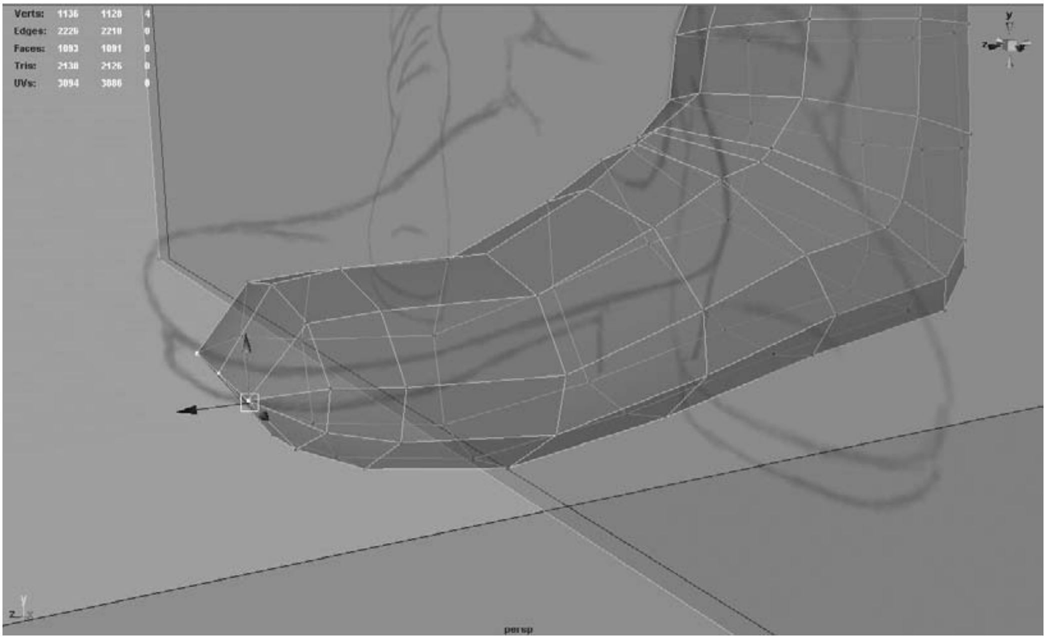


Figure 5.90
Merge the vertices of the toe.

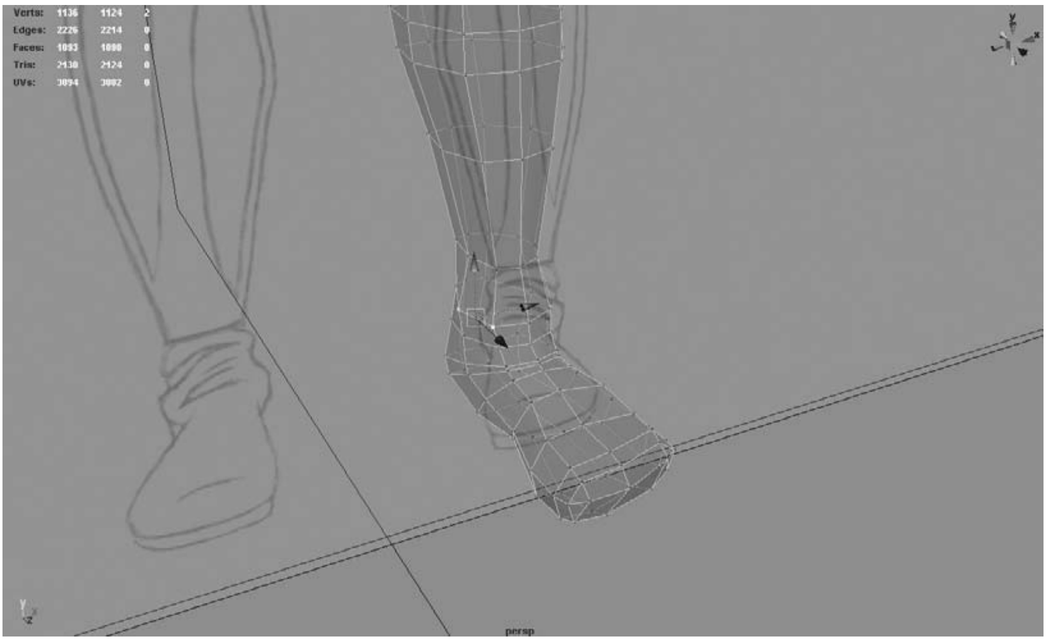


Figure 5.91
Model the shape of the foot.

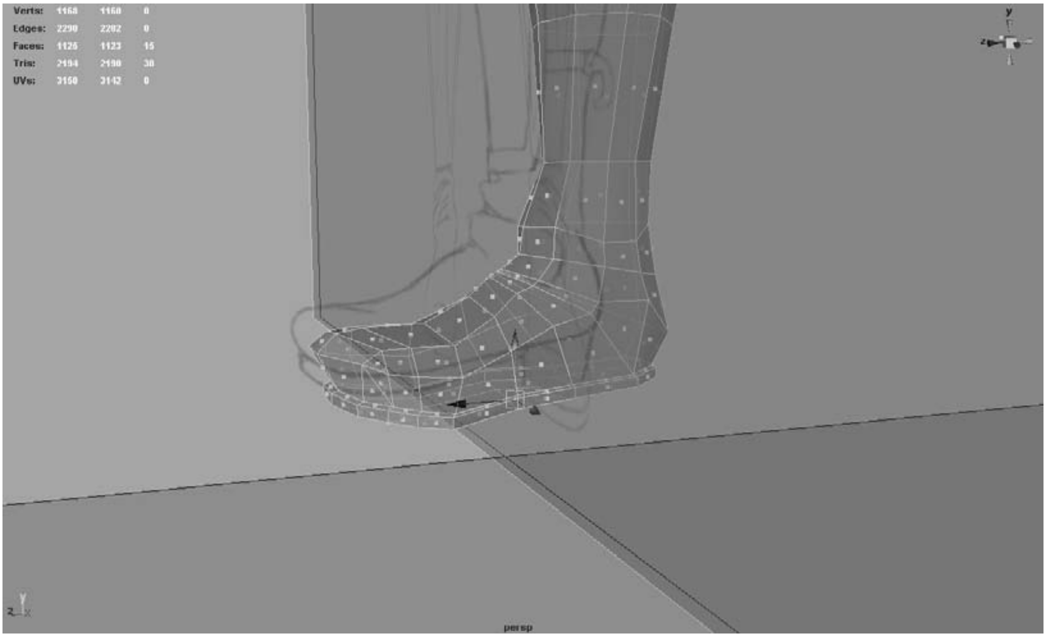


Figure 5.92
Extrude the sole of the boot.

4. Select the faces on the bottom of the boot and extrude them to form the sole, as shown in Figure 5.92.
5. Select the faces at the back of the boot and extrude the heel as shown in Figure 5.93. You will probably have to move the vertices around the heel to get the right look for the heel.
6. Although it isn't part of the foot, there is an element of the model that yet needs to be modeled around the knee: the character's greaves. Switch to front view and move the vertices around the bottom of the knee plate as shown in Figure 5.94.
7. Split the polygons to form a lip from which to extrude the knee plate, as shown in Figure 5.95.
8. Extrude the faces of the split polygons up over the kneecap.
9. Shape the extruded polygons to form the top of the greaves and merge the vertices around the edge, as shown in Figure 5.96.

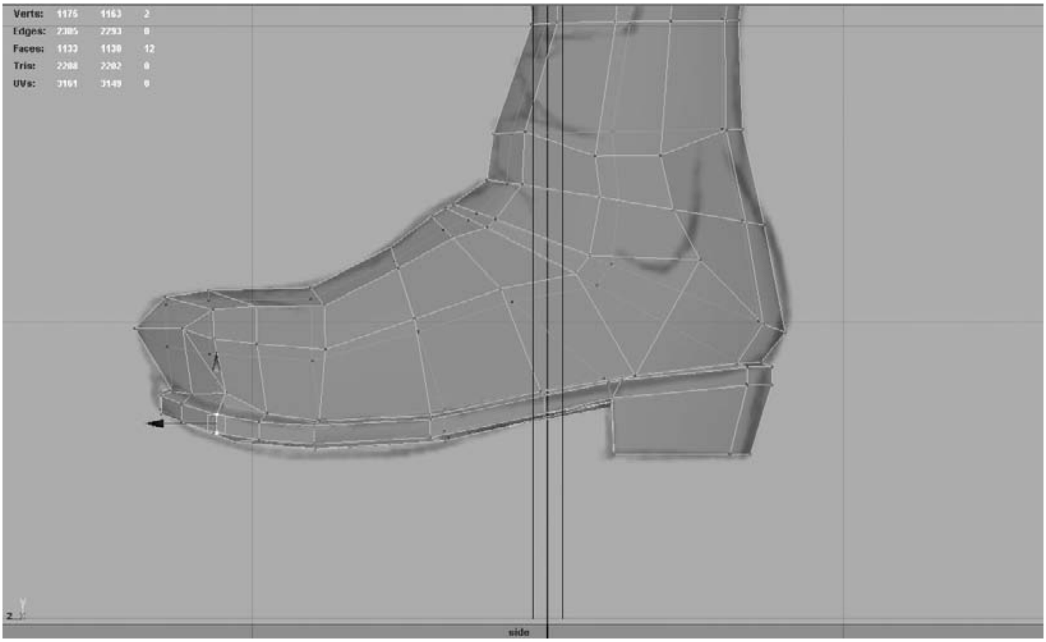


Figure 5.93
Extrude the heel of the boot.

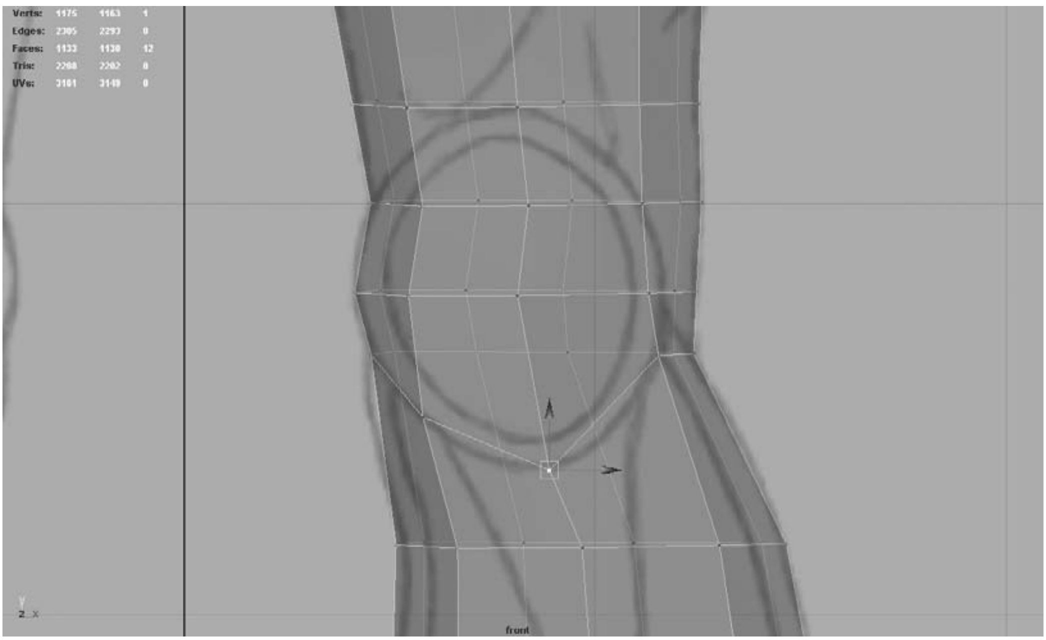


Figure 5.94
Adjust the vertices around the knee to the template.

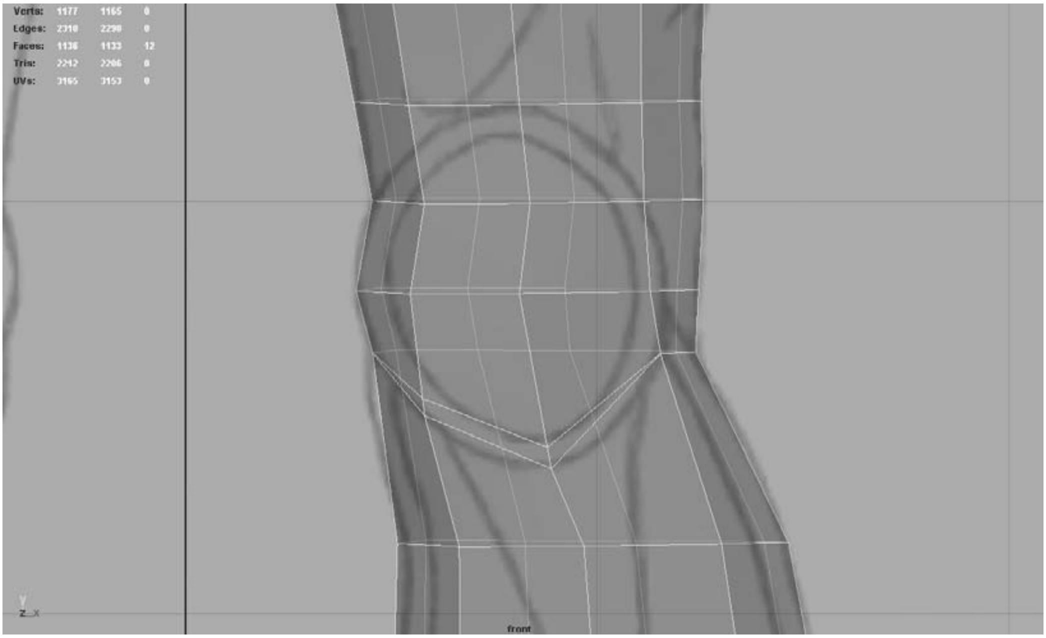


Figure 5.95
Split the polygons near the bottom of the knee plate.

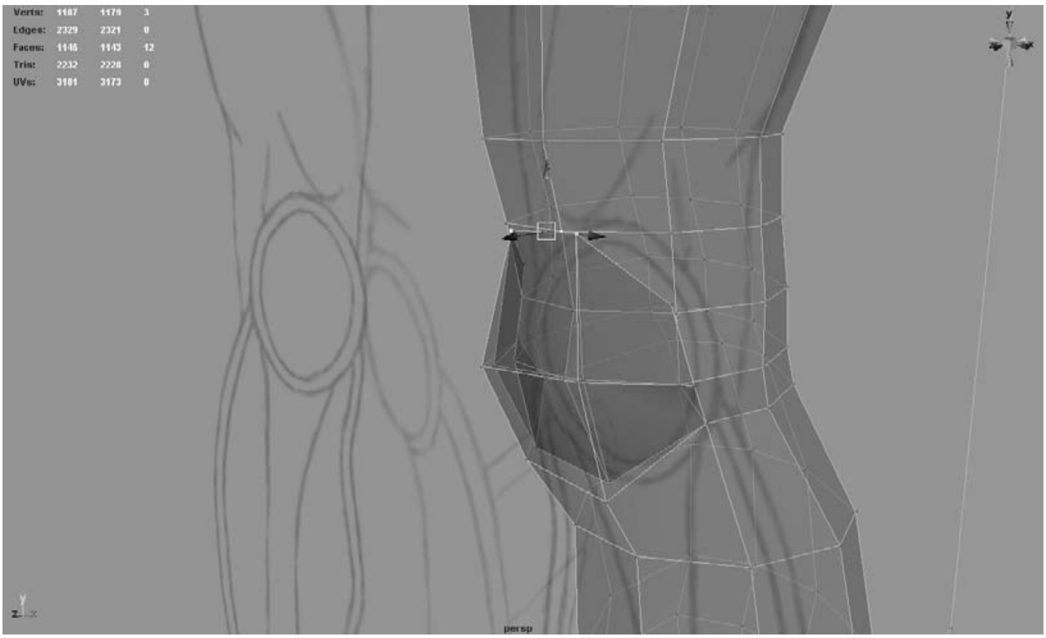


Figure 5.96
Create the knee plate.

The model is almost finished—just a few odds and ends to wrap up before you are through.

Finishing the Model

The basic form of the model is now finished, and he looks almost complete—well, half of him anyway. There are just a few things you need to wrap up to make him ready for texturing and animation.

1. You might have noticed that the shoulder plate is a separate object from the rest of the model. Select it and the character model and combine them as shown in Figure 5.97.
2. You might also have noticed that there is no geometry connecting the shoulder plate with the breast plate. Select the faces on top of the shoulder plate where the hinge is and extrude these faces into the model, as shown in Figure 5.98.
3. Delete the extra polygons along the center line, where the extrusion for the leather straps were, as shown in Figure 5.99.

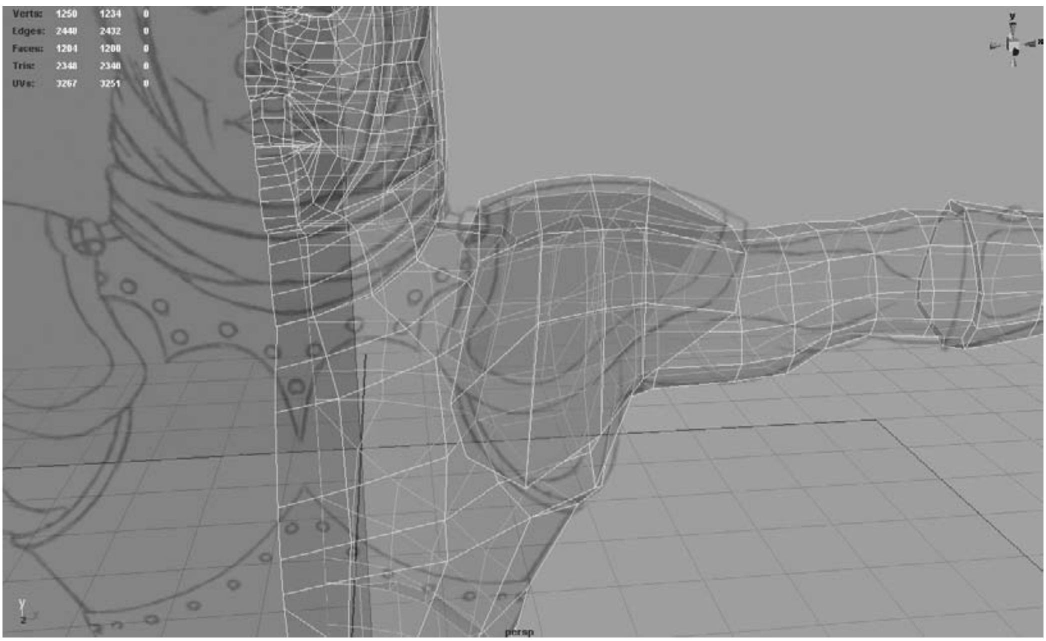


Figure 5.97
Combine the shoulder plate with the rest of the model.

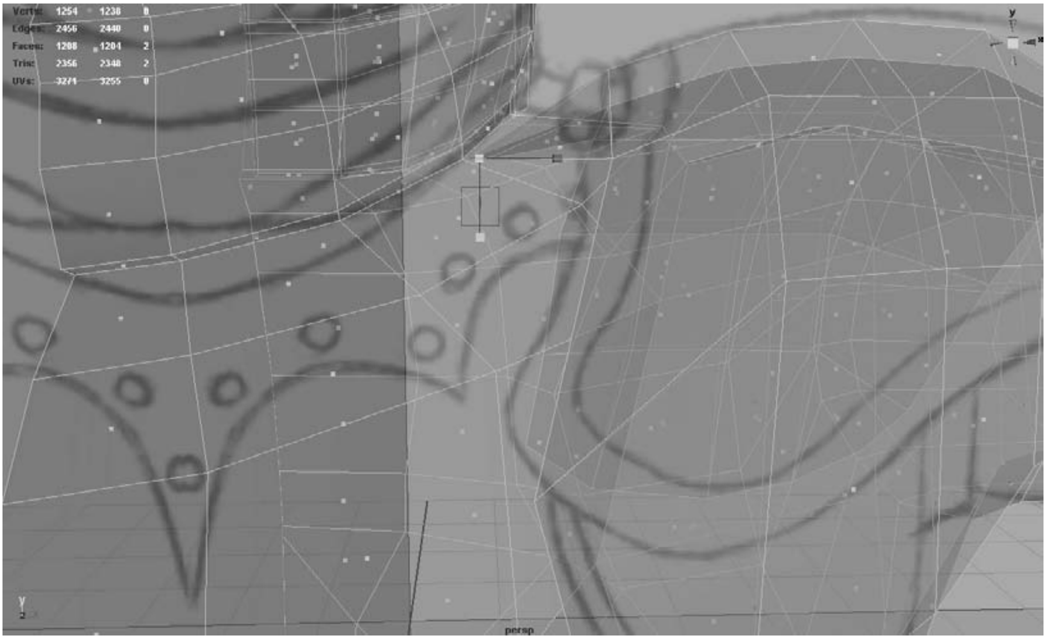


Figure 5.98
Extrude faces from the shoulder plate into the breast plate.

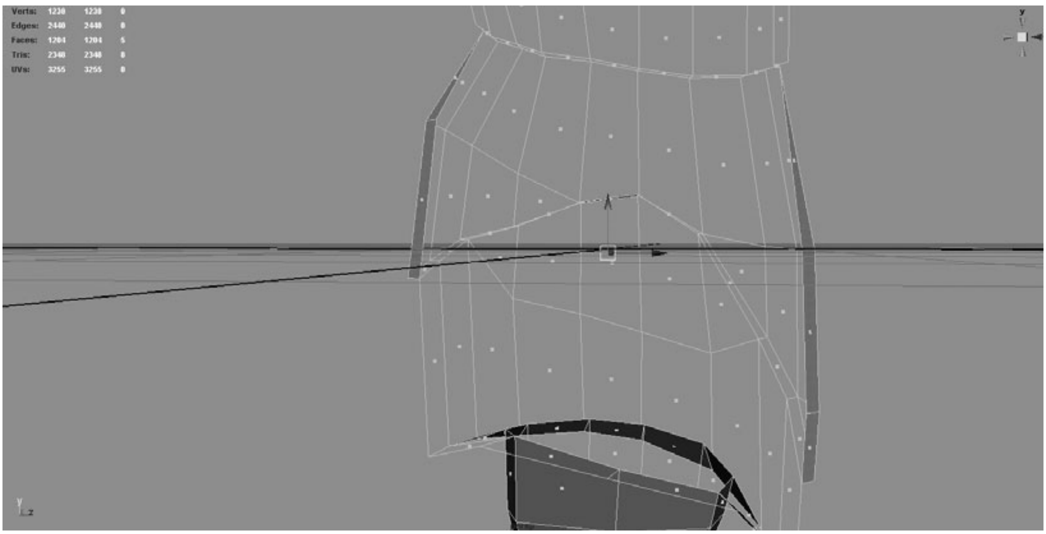


Figure 5.99
Delete unneeded polygons near the hips.

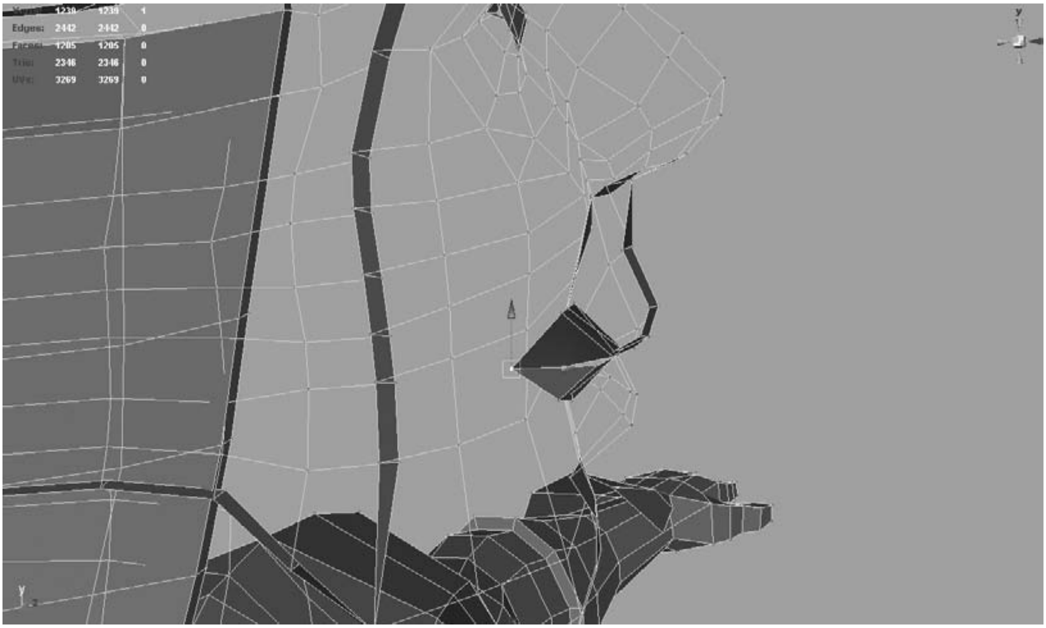


Figure 5.100
Extrude and shape the inside of the mouth.

4. If the character is going to talk or have facial animation, you need to create the inside of his mouth. To begin, select the edges where the lips meet and extrude them into the face, as shown in Figure 5.100.
5. Extrude again, forming the mouth cavity.
6. Merge the vertices at the back of the mouth as shown in Figure 5.101.
7. Select the faces on the bottom part of the back of the mouth and extrude them forward to form the tongue, as shown in Figure 5.102.
8. Create a new polygon pipe object that has a radius of 0.2, a height of 0.25, and a thickness of 0.02. Change the Subdivisions Around Axis setting to 12, the Subdivisions Along Height setting to 1 and the Subdivisions On Cap setting to 1, as shown in Figure 5.103. This object will serve as the bottom teeth.
9. Move the pipe object up to the mouth where the bottom teeth should be.
10. You need only about a quarter of the pipe object, so delete the back faces and the ones that extend past the center line.

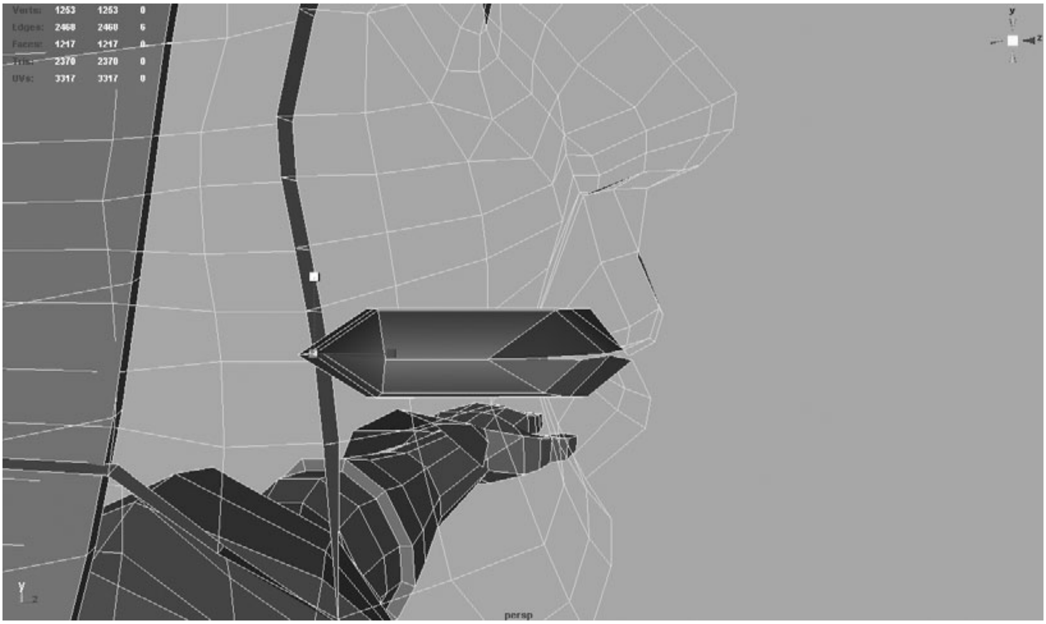


Figure 5.101
Merge the vertices to close the back of the mouth.

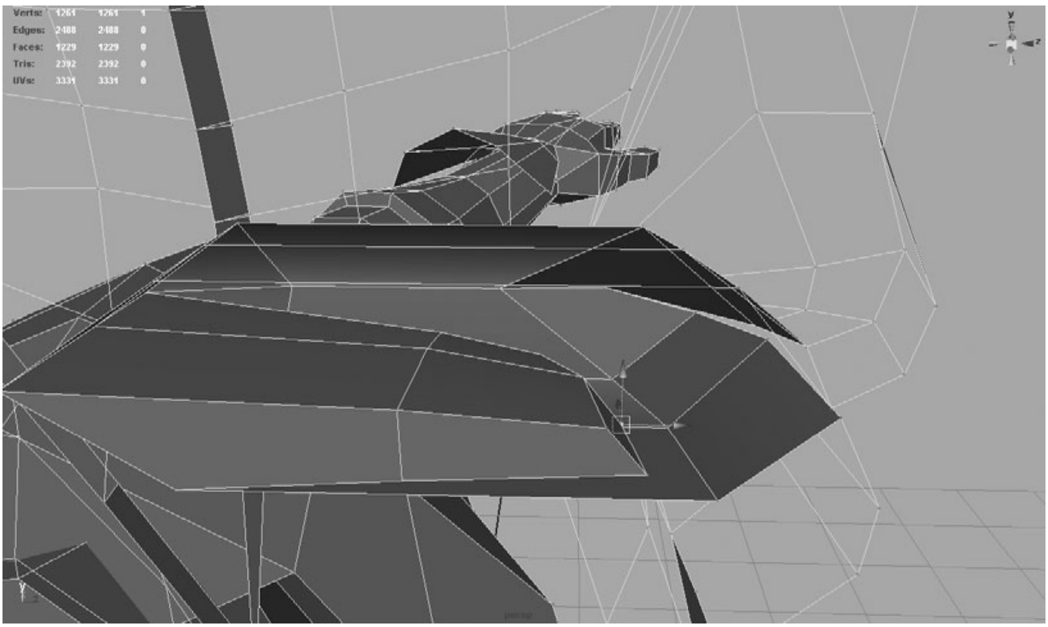


Figure 5.102
Extrude the tongue.

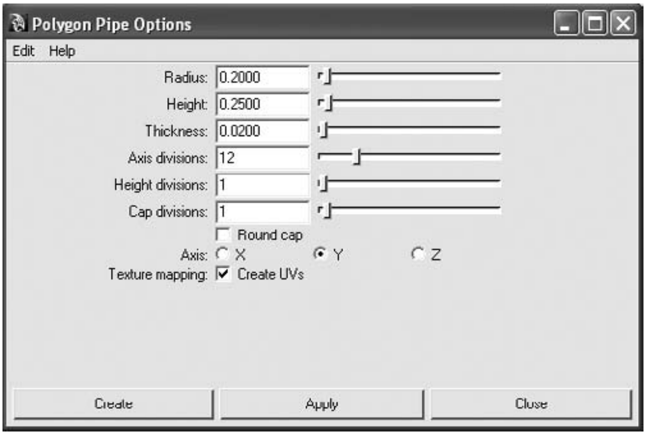


Figure 5.103
Create a new polygon object for the teeth.

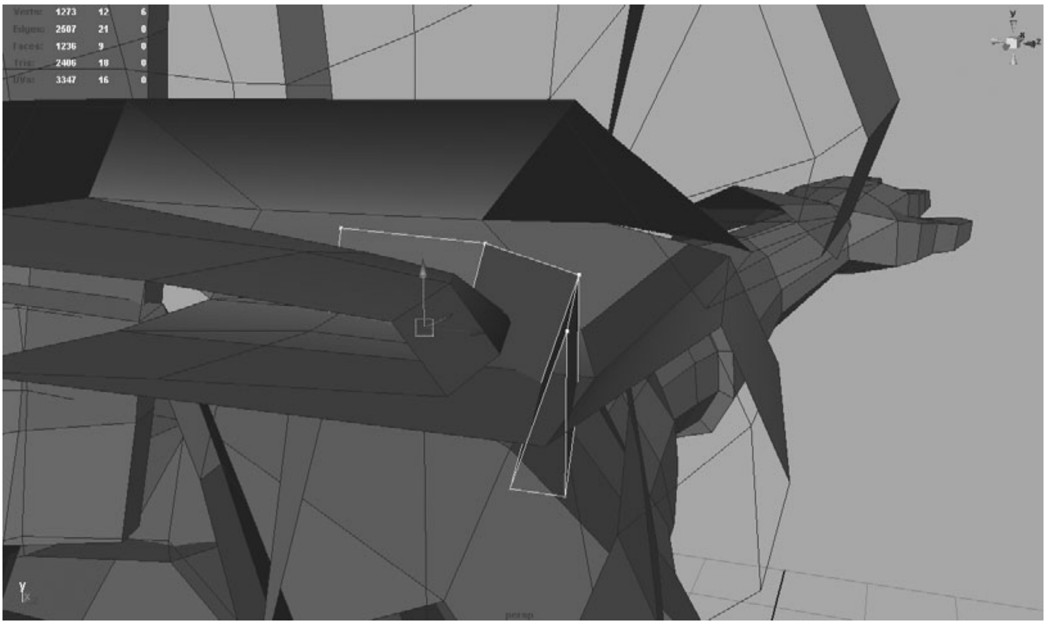


Figure 5.104
Merge the vertices at the top of the teeth.

11. Merge the vertices on the top of the teeth, as shown in Figure 5.104.
12. Rather than creating another pipe object for the top teeth, simply mirror the bottom teeth. Launch the Mirror Geometry function via the Mesh menu and mirror the bottom teeth in the $+y$ axis, as shown in Figure 5.105.

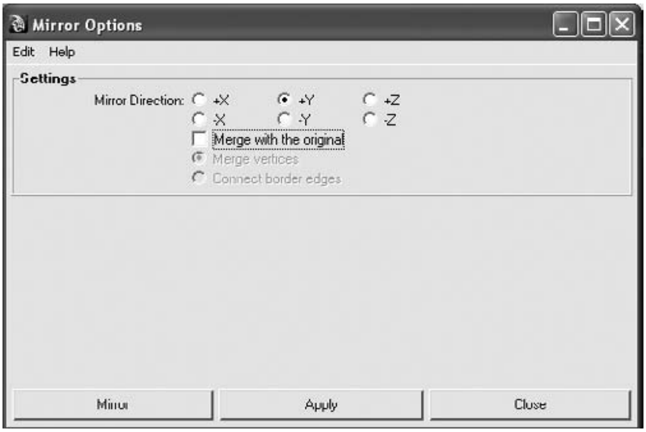


Figure 5.105
Mirror the bottom teeth upward to form the top teeth.

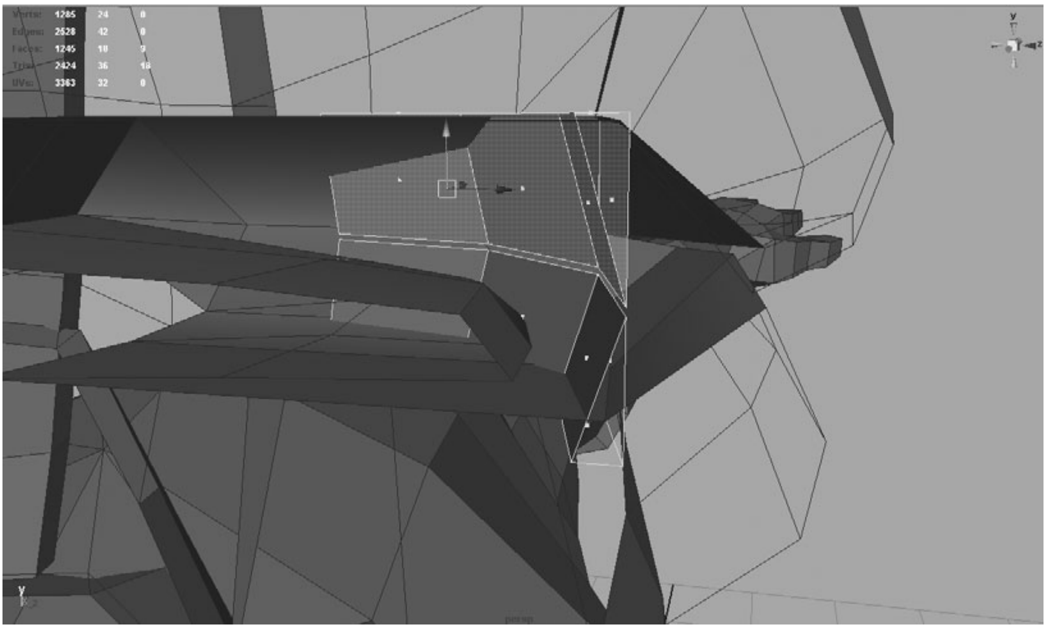


Figure 5.106
Move the top teeth up in the mouth.

13. Select the faces of the top teeth and move them up in the mouth, as shown in Figure 5.106.

14. The character has an eye but no eyeball. To make one, create another polygon sphere object with a radius of .15, change the Subdivisions

Around Axis setting to 8, and change the Subdivisions Along Height setting to 8 in the z axis, as shown in Figure 5.107.

15. Before moving the eyeball into position, you need to create the tear duct on the inside of the eye. Create it by extruding edges and merging vertices as shown in Figure 5.108.

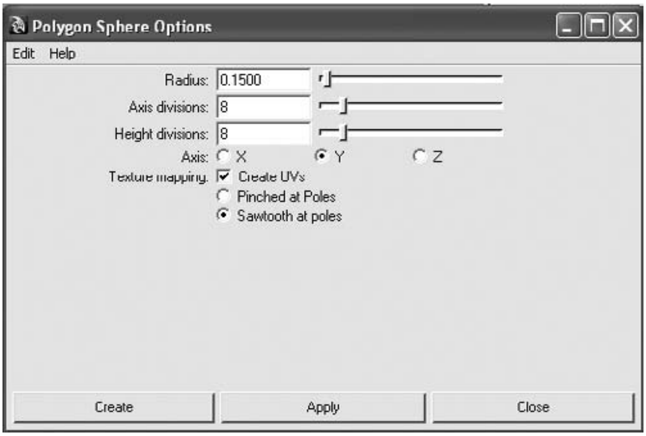


Figure 5.107
Create a polygon sphere for the eye.

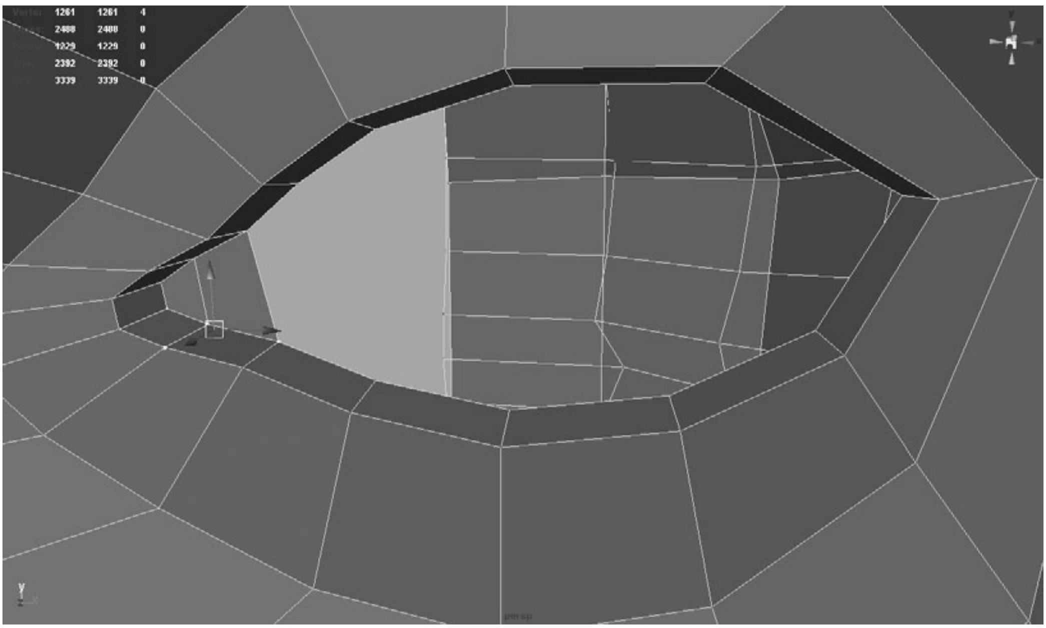


Figure 5.108
Give the character a tear duct on the inside of the eye.

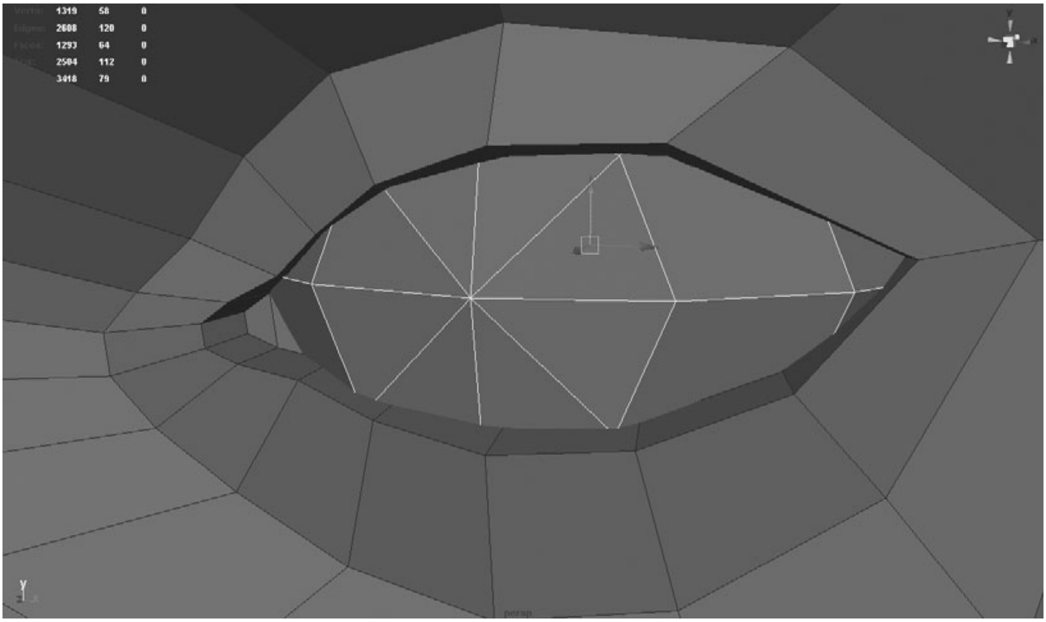


Figure 5.109
Move the eyeball into position.

16. Move the eyeball into position, as shown in Figure 5.109. (You may need to make adjustments to the eyelids so that the eye fits correctly.)
17. You don't need all the polygons of the eyeball, so select and delete the unneeded ones (see Figure 5.110).
18. The eyeball is the only part of the model that will remain a separate object. This is so that it will be easier to animate the eyes later. The teeth, on the other hand, can be combined with the rest of the mesh, as shown in Figure 5.111.

The model is now finished and needs only to be mirrored to be complete.

Tip

Now might be a good time to save your work. In fact, you should get in the habit of saving your work often. There are not many things more frustrating than losing a day's work because of a system crash or some other unfortunate event!

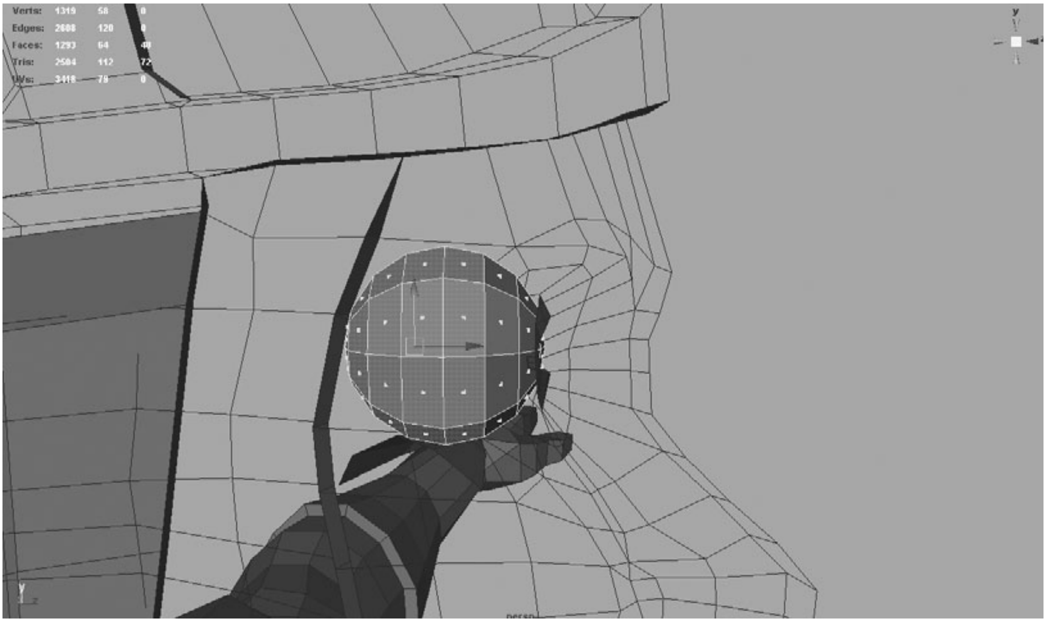


Figure 5.110
Delete the polygons on the back of the eye.

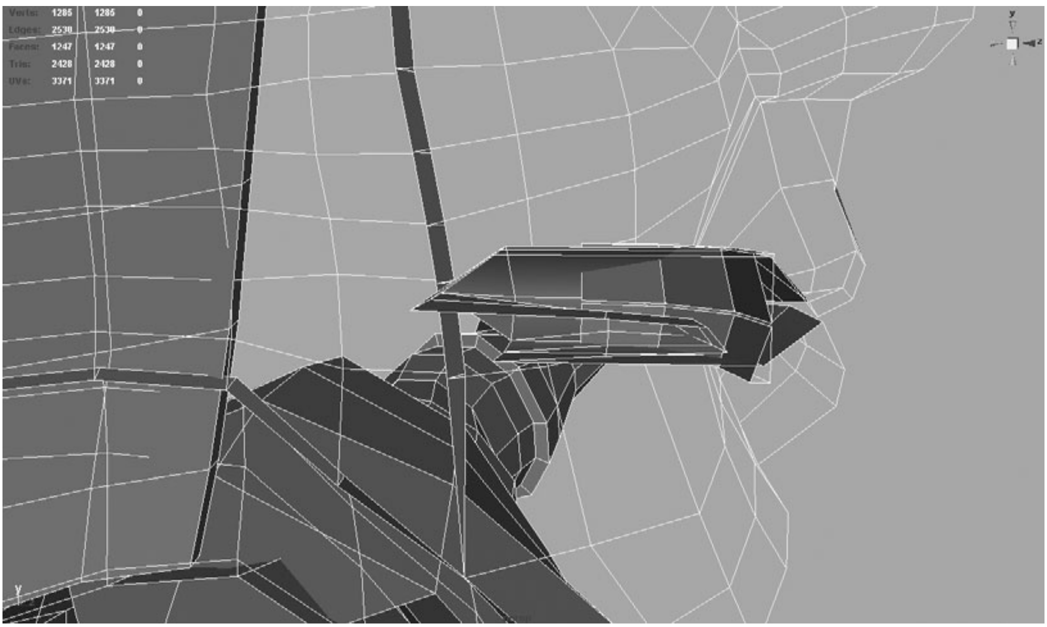


Figure 5.111
Combine the teeth with the rest of the model.

Mirroring the Model

The mirroring process is a great way to save a ton of work when creating symmetrical models. I even use it for models that are only *partially* symmetrical because mirroring makes those parts that are symmetrical perfect, allowing me to simply adjust the model for the asymmetrical parts after the mirroring process.

Before you bring up the Mirror tool, you can save a lot of headaches by making sure the seam along the center line is perfect. To do so, zoom in on the center line from the front view and snap any stray vertices back to the center line. (To avoid messing up your work, only snap them in the x axis.) When you are sure the seams are perfect, you can proceed to the mirroring process

Launch the Mirror tool and mirror the model in the −x axis, as shown in Figure 5.112. If your vertices are correctly aligned along the center line, there

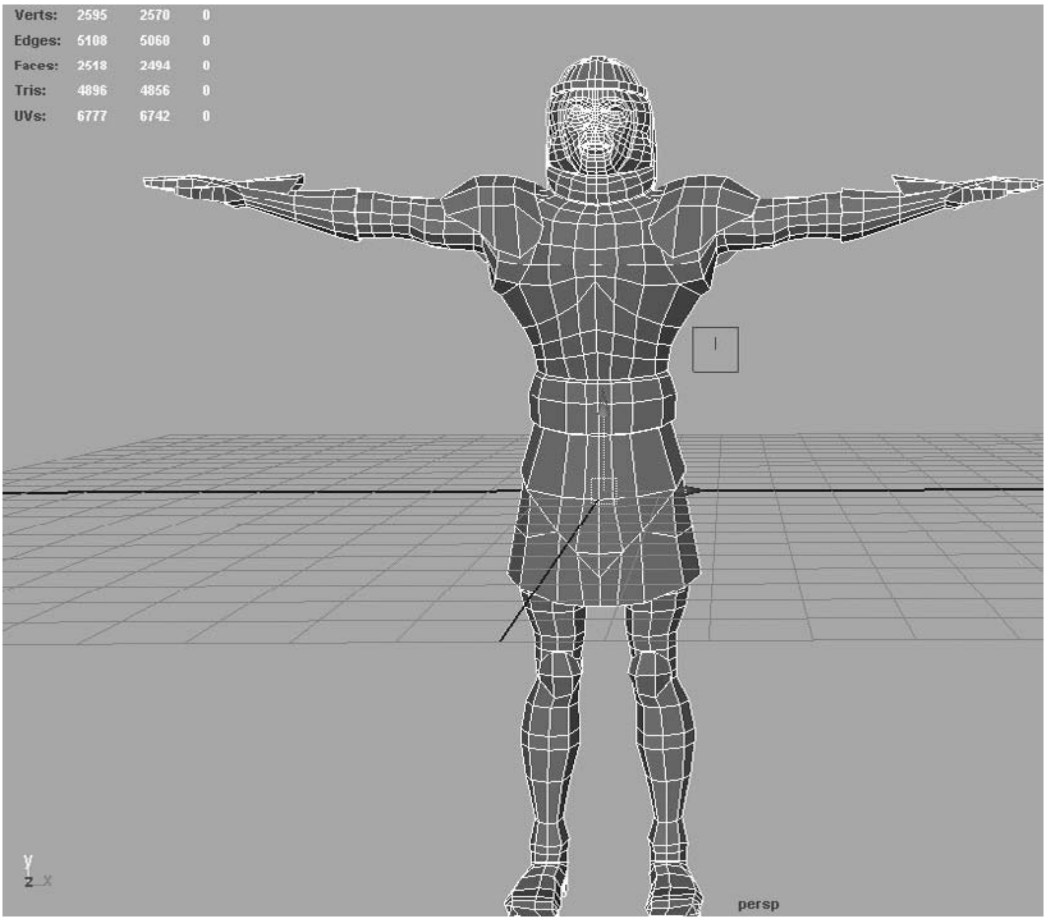


Figure 5.112
Mirror the model.

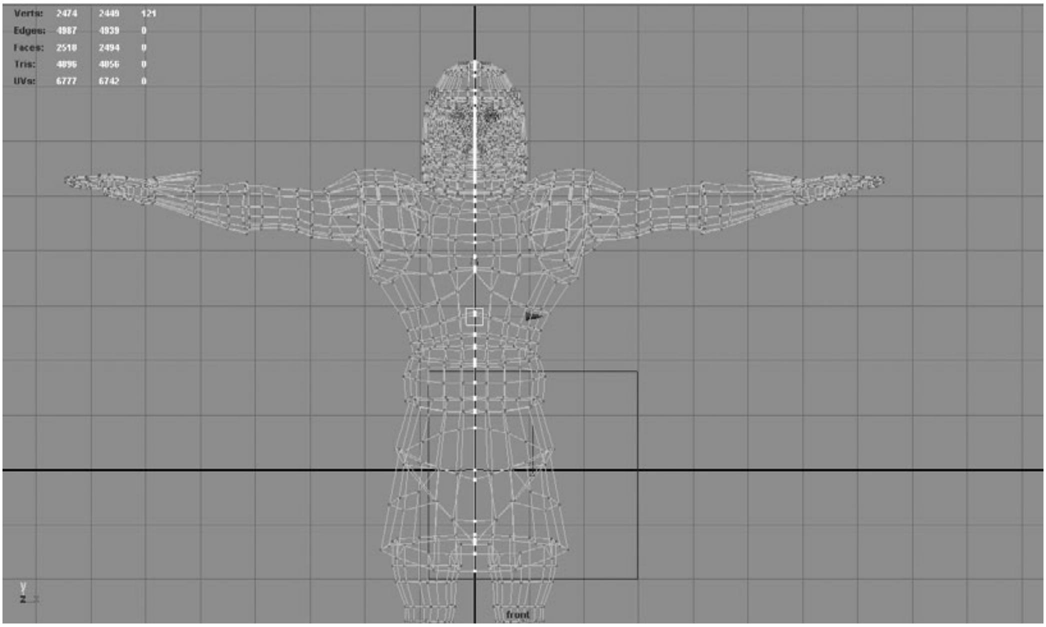


Figure 5.113
Merge the vertices along the center line.

should be no gaps between the two halves. If there are gaps, you need to undo the Mirror operation and fix the problem, and then go back and mirror the model again.

I usually mirror objects with the Merge Vertices function turned off because that feature sometimes merges vertices that I don't want merged. Instead, after the mirror process, I select all the vertices along the center line and merge them as a separate function, as shown in Figure 5.113.

Your character currently has only one eye because the eyeball was a separate object and was not mirrored with the rest of the model. To add the other eye, first select the eyeball and duplicate it. Then, using the Channel box, change its X translation from positive to negative, as shown in Figure 5.114, to place it correctly.

There you have it. As shown in Figure 5.115, the model is now finished and ready to texture. Because this chapter is already long enough, I'll wait for the next one to talk about materials and textures for your character models. In the meantime, you can admire your modeling work.

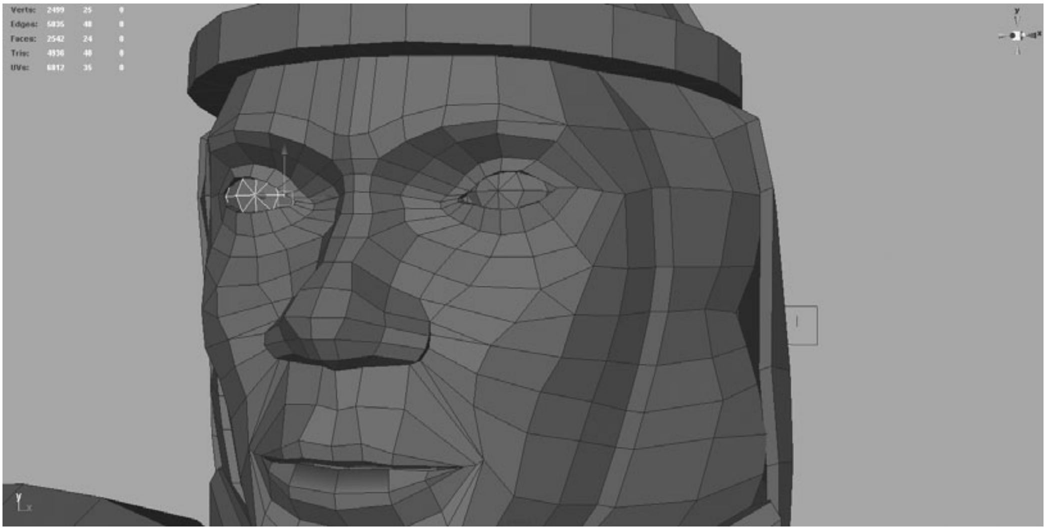


Figure 5.114
Put a duplicate eyeball in the empty eye socket.

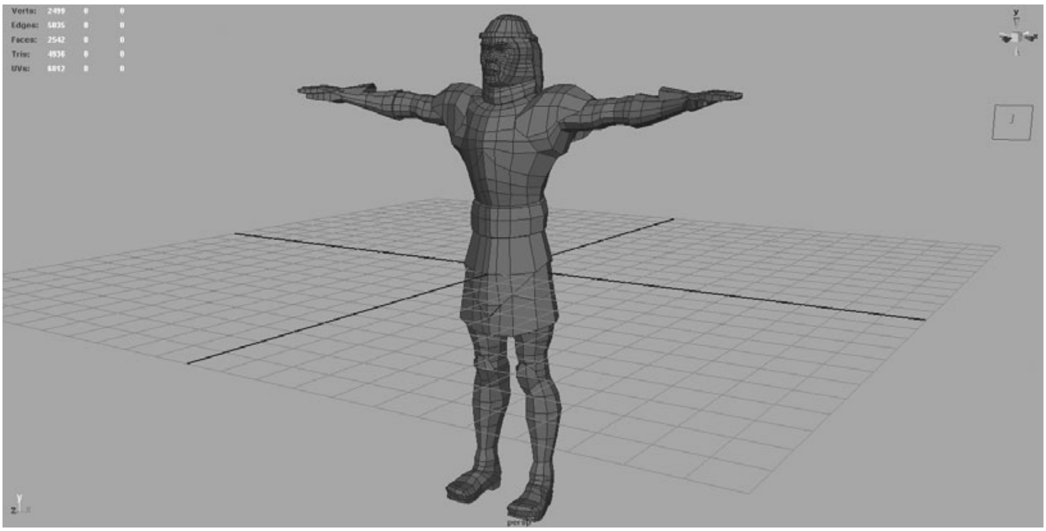


Figure 5.115
The model is ready for texturing.

Summary

You covered a lot of ground in this chapter, including the following:

- Setting up the template
- Modeling the face

- Modeling the head
- Modeling the breast plate
- Modeling the arm
- Modeling the hand
- Modeling the leg
- Modeling the foot
- Finishing the model
- Mirroring the model

CHAPTER 6

Texturing the Character

If you could use an unlimited number of polygons when modeling your game characters, you could conceivably model all of the detail. Modeling all the detail in even a simple character, however, would require polygon counts in the hundreds of thousands—too much for a game engine to handle efficiently. For this reason, most game characters are limited to 5,000 to 10,000 polygons at the most or as few as 500 to 1,000 polygons, depending on the platform. Indeed, I have worked on games in which the individual characters could have no more than 200 polygons each. It is pretty obvious from these numbers that game characters must show detail in some other way. To add detail to a game character without inflating the number of polygons, texture maps are used. A *texture map* is a 2D image that is attached to the UVs of a model, adding detail, color, and texture to the model.

In this chapter, you will create textures for the model you built in the previous chapter. You will concern yourself primarily with the color map of the character, because the color map is universal to game engines. Some games engines support other maps like bump maps and specular maps, but many do not. A *bump map* is used to simulate light hitting an uneven surface. A *specular map* is used to vary the specularity of a surface (in other words, the how shiny the object appears). As game engines become more powerful, these types of maps will become more common.

Preparing the Model

Before you can apply a texture to your model, you must organize the model's UVs. *UV* is the term Maya uses to describe how the pixels of a texture are related to the vertices of a model. When a model is created in Maya, it contains a default set of UVs for the original primitive. Those UVs, however, are often unusable in the final construction of the model because of the many changes made to create a character. That means the UVs for the model must be reorganized so they make sense for creating textures. For example, Figure 6.1 shows the UVs of the character you just created in the UV editor to the left of the model. Notice that the UVs are superimposed over each other, causing many to be overlapped and jumbled. It would be impossible to create a texture map that would work well with this arrangement of UVs.

Note

Notice that I only have half of the model present in Figure 6.1. That's because, as with modeling, it is much easier when creating a symmetrical character to arrange the UVs for only half the model and then mirror the model later.

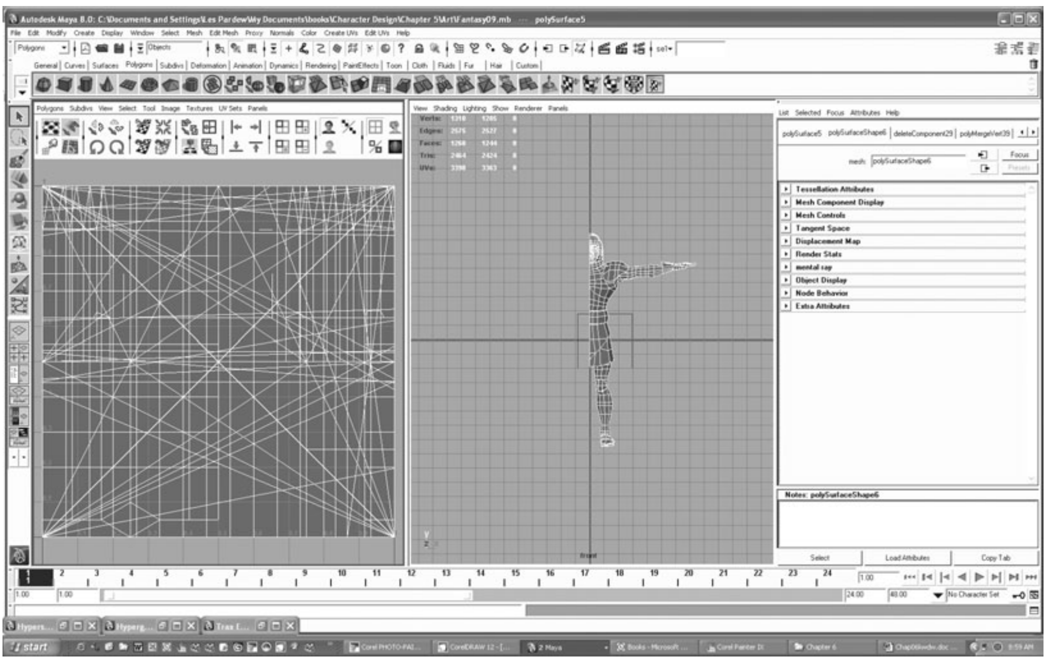


Figure 6.1
The UVs of the character are overlapping and jumbled.

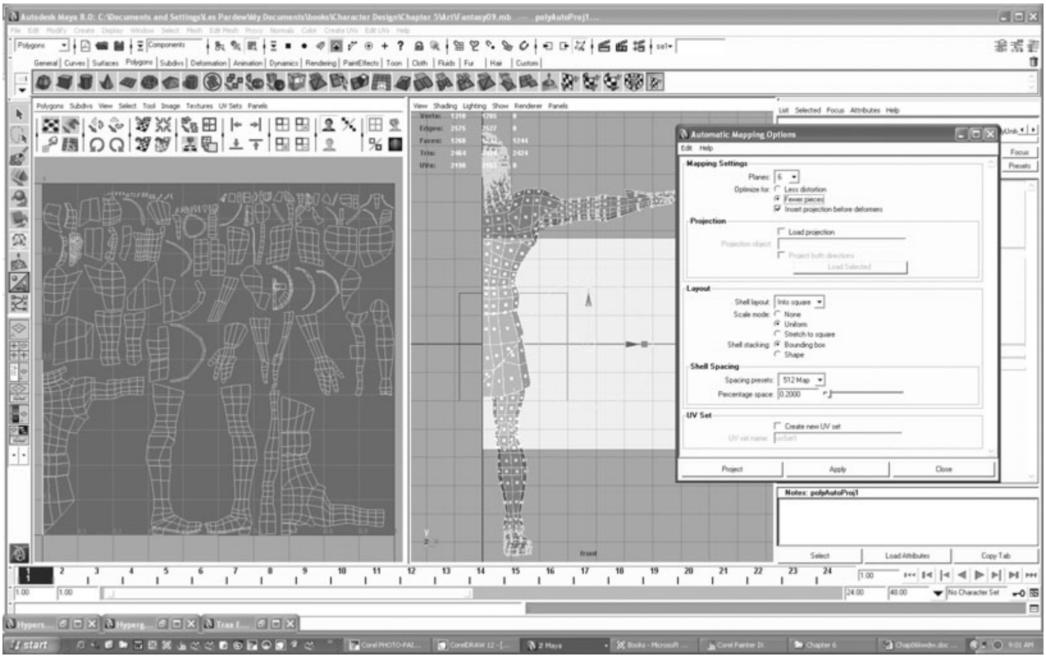


Figure 6.2
Automatic Mapping was used to organize the UVs of the character.

Mapping UVs

Maya offers a number of ways to organize UVs on a polygon mesh, but the one I discuss here is the Automatic Mapping feature Found in the Create UVs menu. I like this feature because it quickly organizes the UVs into discernable units that I can then adjust to my liking in the UV editor. Figure 6.2 shows the model after Automatic Mapping has been used. Notice how much better the UVs are organized in the UV editor.

If you wanted to, you could use the organized UVs as they are currently mapped, because Automatic Mapping eliminates all overlapping UVs. The problem is, with Automatic Mapping, the individual clusters of UVs are often small and have many seams. To get a good UV layout, you need to do some hand adjusting of the UVs. In a lot of ways, organizing an Automatic Mapped UV layout into something that can be used for texturing a character is like putting together a puzzle.

1. Start with the character's face. Using the Move and Scale tools, move the face UVs to one side, as shown in Figure 6.3. (You must be in UV Selection mode to move UVs.)

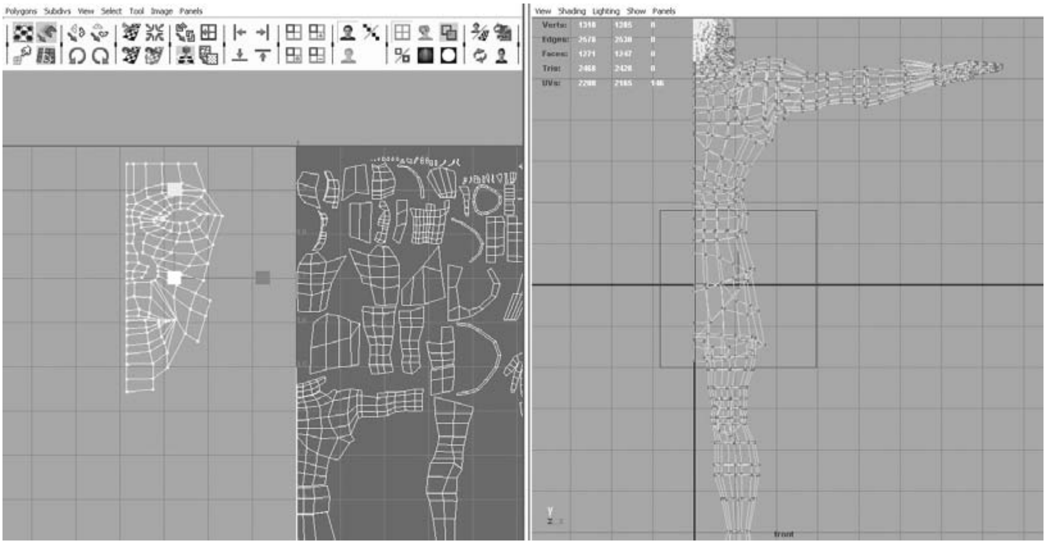


Figure 6.3
Move the UVs of the face to one side.

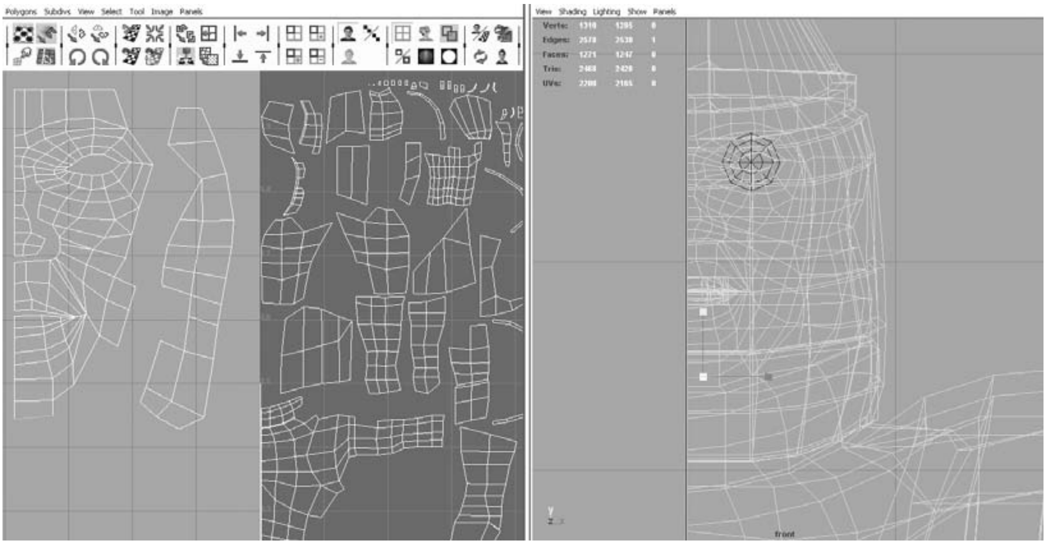


Figure 6.4
Also move the UVs for the side of the face.

Note

The manipulator tools can be used in the UV editor similarly to how they are used in the modeling area. The Move tool moves selected items, Scale expands and shrinks selected items, and Rotate rotates selected items. The main difference in the manipulator functions is that in the UV editor, all functions are in two dimensions instead of three.

2. Find the UVs to the side of the character's face and move them to where the face UVs are, as shown in Figure 6.4.

Note

When you are in Edge selection mode, selecting an outside edge of a group of UVs also selects the corresponding edge. By using this feature, you can check your edges to make sure you fit the correct edges together. In this way, I was able to quickly locate the UVs that are part of the character's face and move them to the side.

Putting the Pieces Together

After you have collected the necessary pieces for the face, you are ready to start putting them together. Fortunately, Maya offers a set of specialized tools to help you with this task. The following shows some of the more common tools and what they do:

Tool Icon	Tool Name	Description
	Flip UVs	Flips UVs in the U direction
	Flip UVs	Flips UVs in the V direction
	Rotate UVs	Rotates UVs in a counter-clockwise fashion
	Rotate UVs	Rotates UVs in a clockwise fashion
	Cut UVs	Separates UVs along selected edges
	Move and Sew UVs	Moves and sews selected edges together
	Sew UVs	Sews selected edges or UVs together
	Split Selected UVs	Separates UVs

3. Select the edges along the face. Notice that their corresponding edges are selected, as shown in Figure 6.5. Make sure that you select only common edges between the two UV groups.

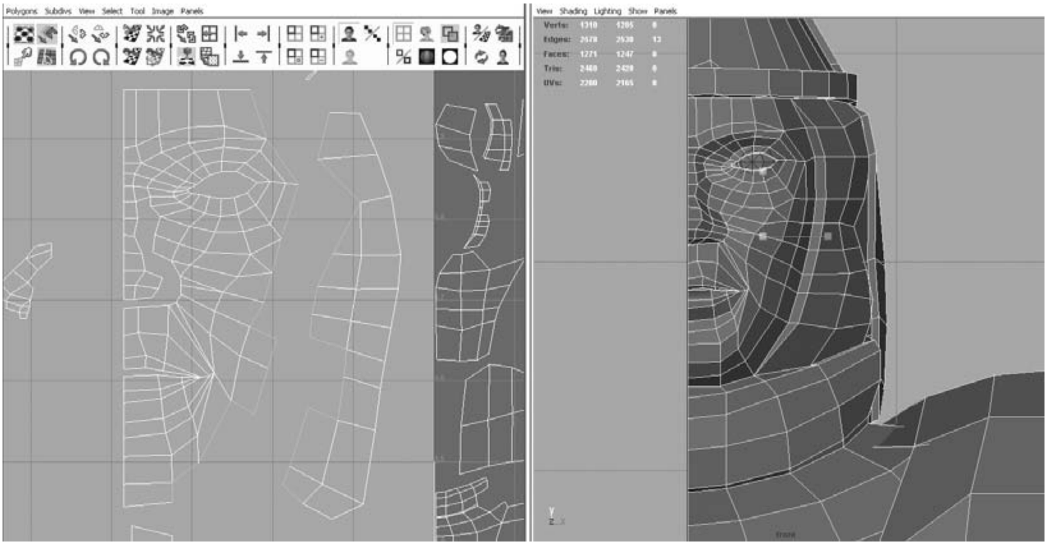


Figure 6.5
Select common edges between UV groups.

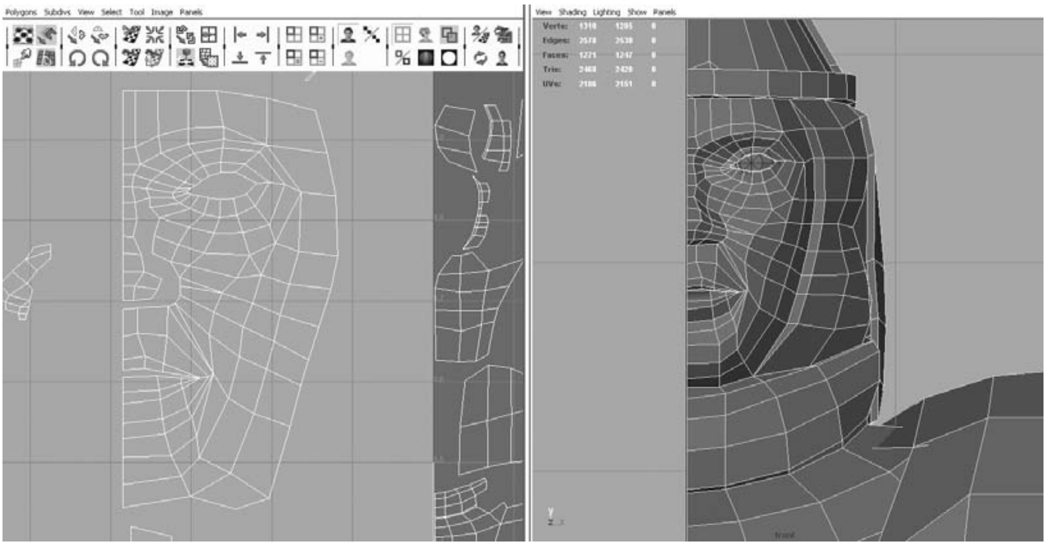


Figure 6.6
Move and sew the selected edges.

4. Move and sew the edges together, as shown in Figure 6.6, using the Move and Sew UVs function.
5. Continue adding pieces to your puzzle. For now, just gather all the UVs that are part of the face.

6. When you get to the eyes, you will need to adjust the UVs to follow the shape of the eye. Use Figure 6.7 as an example of how the UVs around the eye should be moved.

7. Part of my model's lower lip was missing. I found it by selecting the edge and then looking for a match. Figure 6.8 shows where I found the missing piece.

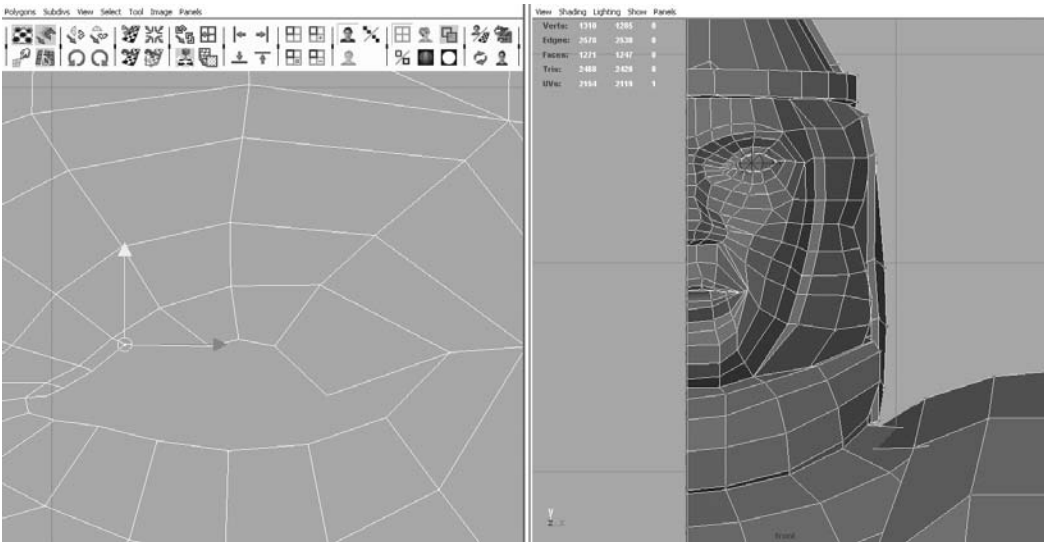


Figure 6.7
Move the UVs around the eye into their correct position.

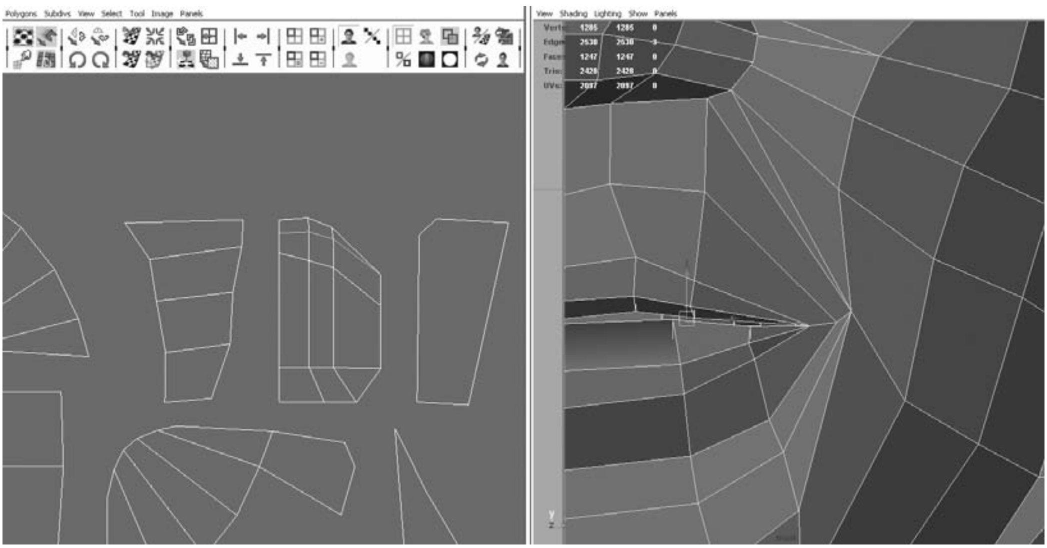


Figure 6.8
Part of the lower lip was missing.

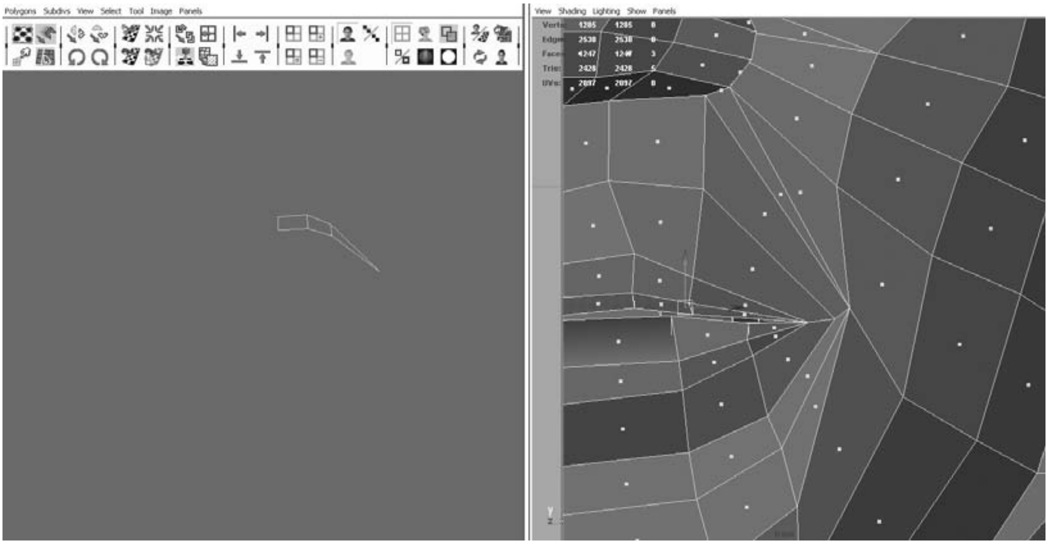


Figure 6.9
Non-selected faces disappear.

You may have a few missing pieces of your model's face as well; make sure to get all of them.

8. I used the Cut UVs tool to separate the missing piece, and then I changed the selection mode to Faces to select the missing piece. When you select faces, all the non-selected faces disappear, as shown in Figure 6.9. This feature is very useful for picking up UVs that were recently cut from a group; you can then change back to UV selection mode and grab only the UVs of the selected faces.

9. Move the selected UVs away from the UVs from which they were cut, as shown in Figure 6.10. This method of selecting UVs helps protect the UV groups from the major distortion caused by moving individual UVs to separate UV groups.

10. Use the Move and Sew UVs function to attach the UVs to the mouth, as shown in Figure 6.11.

11. Continue to gather the UVs of the face until they are all in one group. When you are finished, your group should look similar to the one in Figure 6.12.

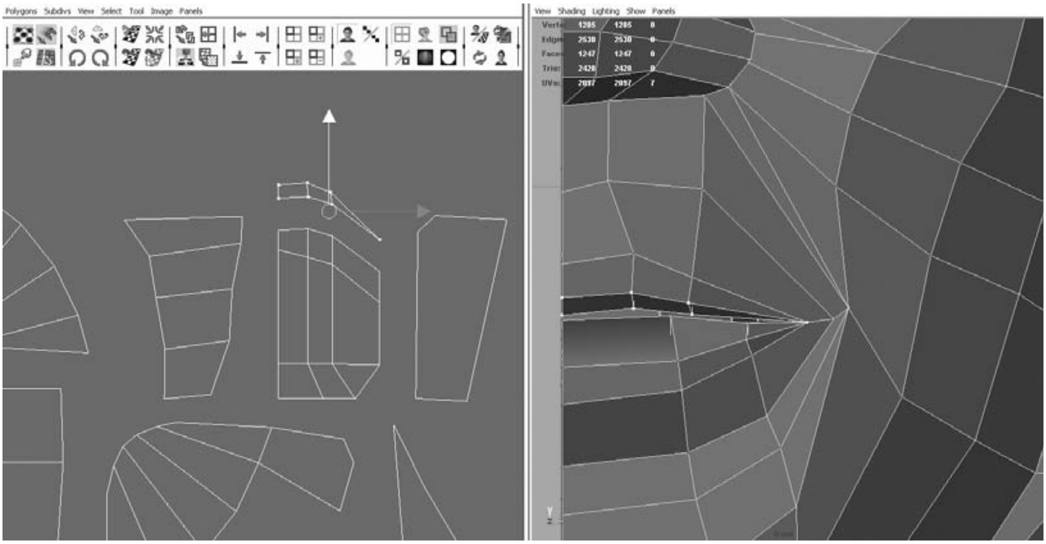


Figure 6.10
Move the selected UVs away from the group from which they were just cut.

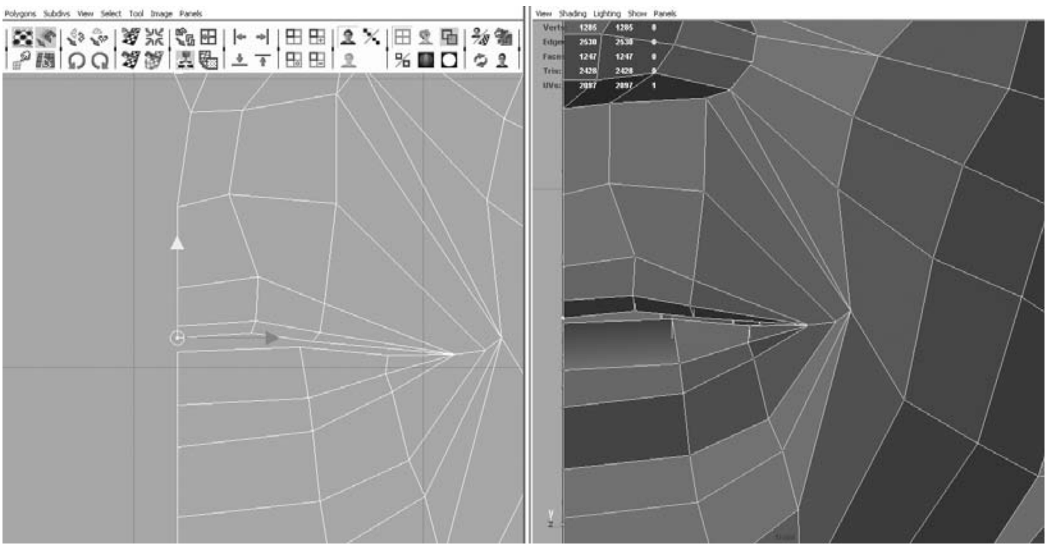


Figure 6.11
Use the Move and Sew UVs function to attach the missing part of the lip to the face group.

12. Now that you have one section of the model's UVs organized, move that section to a location by itself and start working on the head wrap, as shown in Figure 6.13. Notice that I am isolating UV groups by section on the model. This is because there is a natural seam between the face and the scarf.

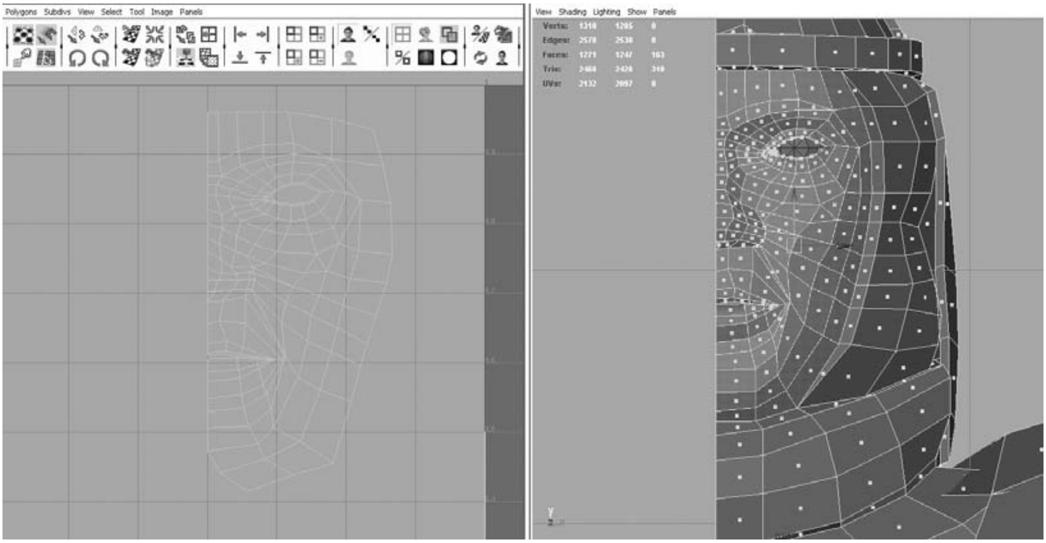


Figure 6.12
Combine all the UVs of the face into one group.

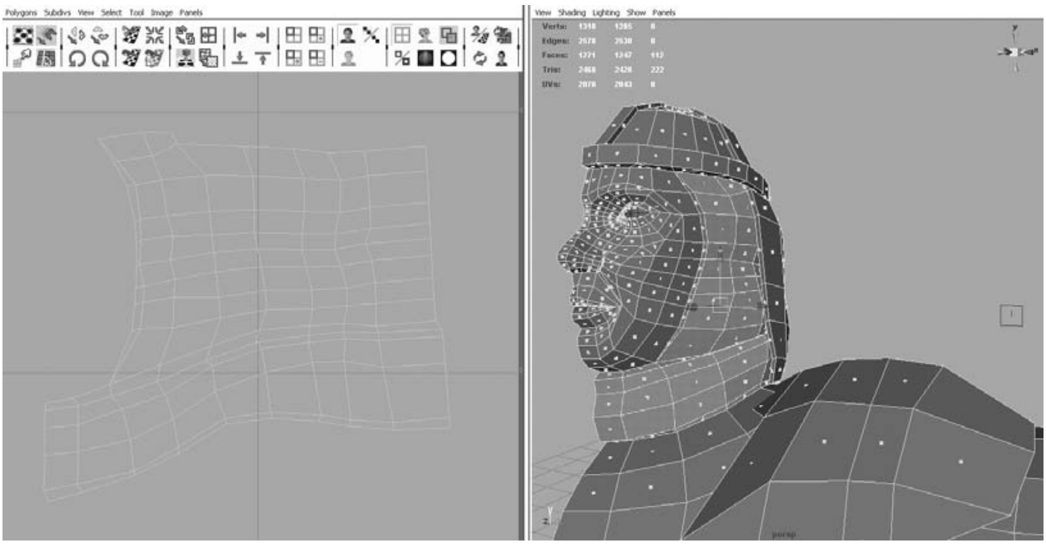


Figure 6.13
Next, work on the character's head wrap.

By creating UV groups with boundaries on natural seams, you make the painting of the texture maps much easier later on.

13. From the head wrap, move on to the helmet. Figure 6.14 shows how it is put together; as you can see, there are four pieces.

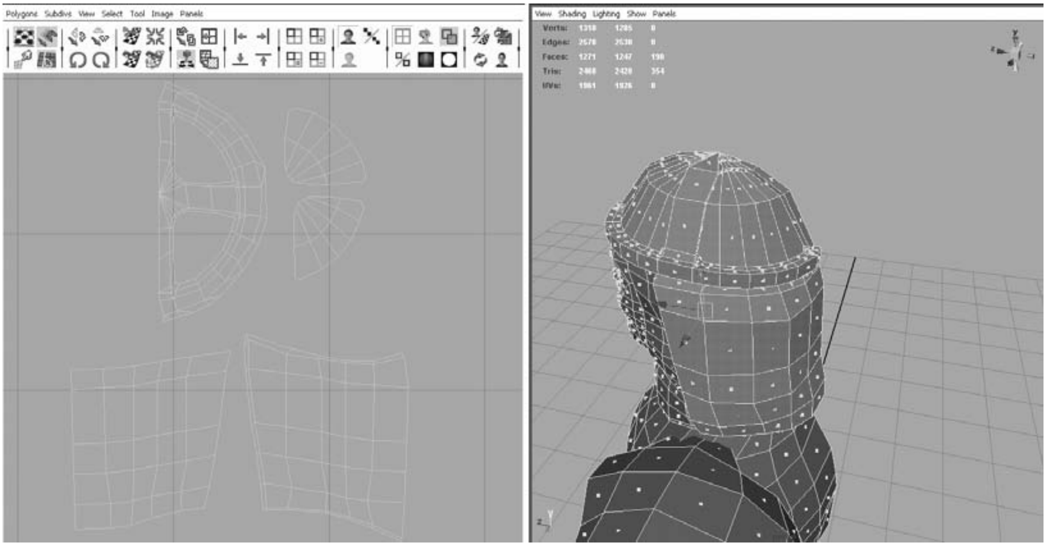


Figure 6.14
The helmet is divided into four groups.

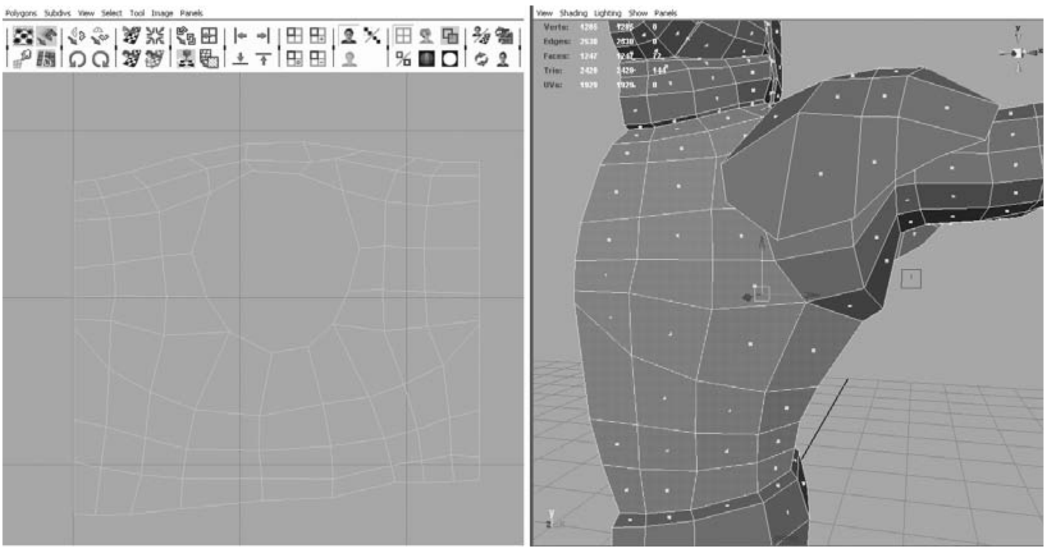


Figure 6.15
Organize the UVs of the Breast plate.

14. Continue organizing the UVs by next working on the breast plate, as shown in Figure 6.15.
15. Organize the leather belt and straps, as shown in Figure 6.16.
16. Put together the UVs of the chainmail tunic, as shown in Figure 6.17.

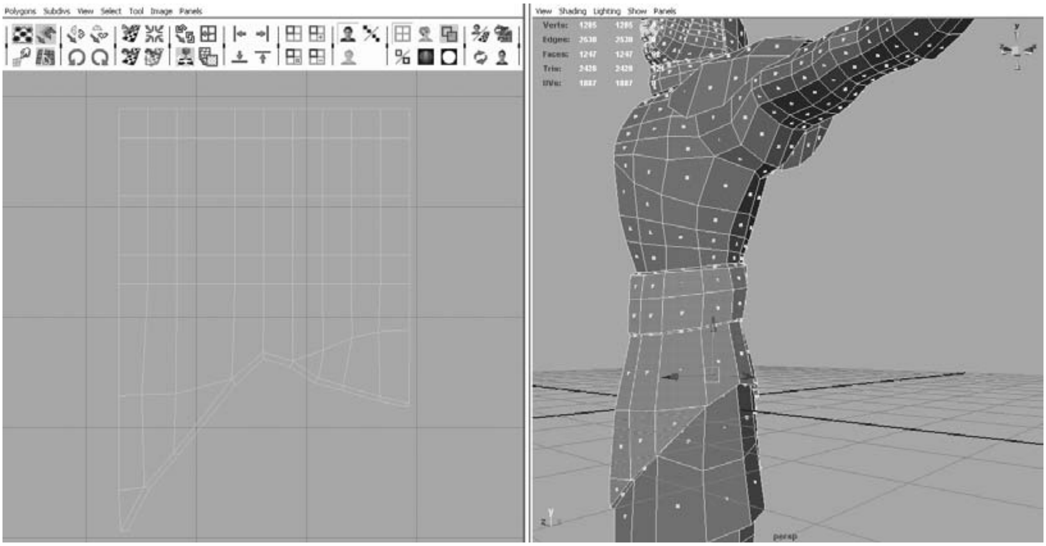


Figure 6.16
Work on the belt and straps.

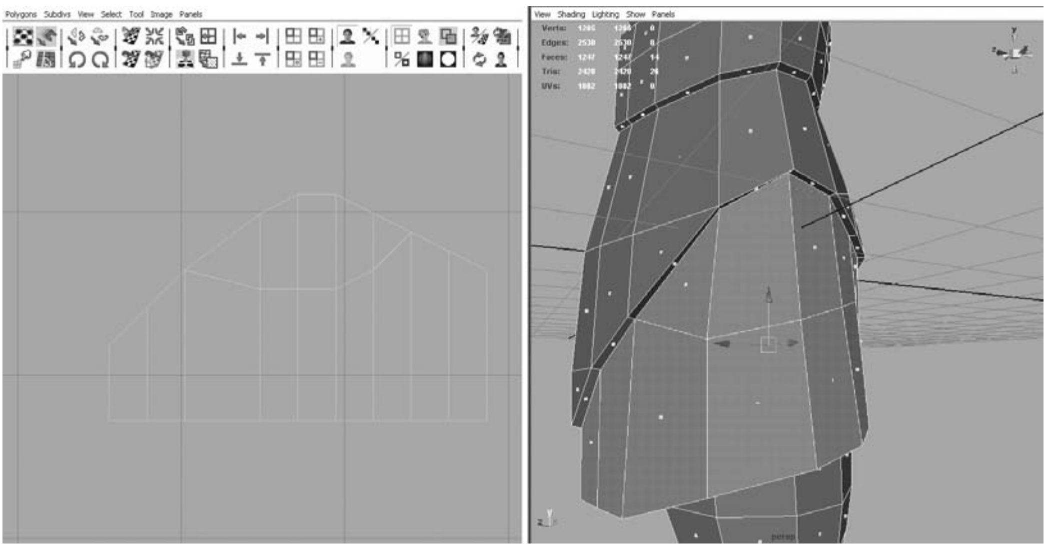


Figure 6.17
Work on the bottom of the chainmail tunic next.

17. The UVs of the shoulder plate are next, and should look like those shown in Figure 6.18.
18. Continue outward by isolating and organizing the UVs that make up the chainmail covering the character's arm (see Figure 6.19).

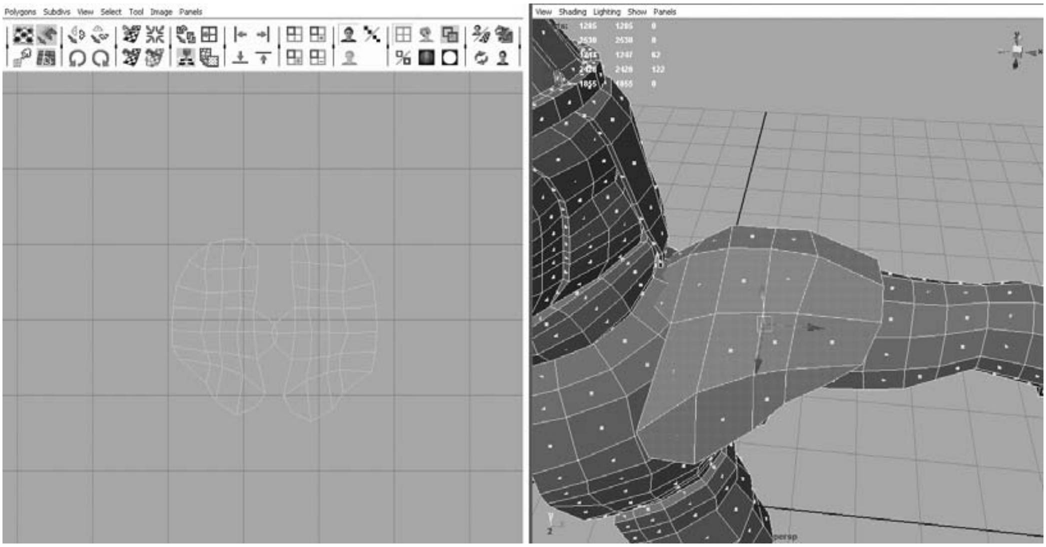


Figure 6.18
Organize the UVs of the shoulder plate.

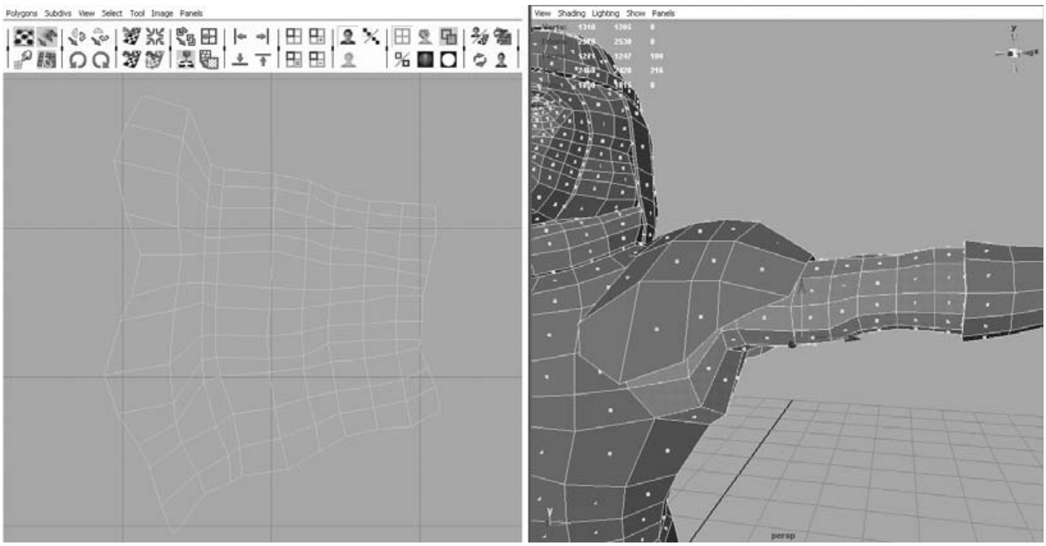


Figure 6.19
Organize the UVs of the character's arm.

19. Move down the arm to the metal bracers, as shown in Figure 6.20.
20. The gantlet is next, and it may be a little more complicated to organize; see how I placed the UVs in Figure 6.21 for help.
21. The UVs between the leg and the bottom of the tunic also need to be organized, as shown in Figure 6.22.

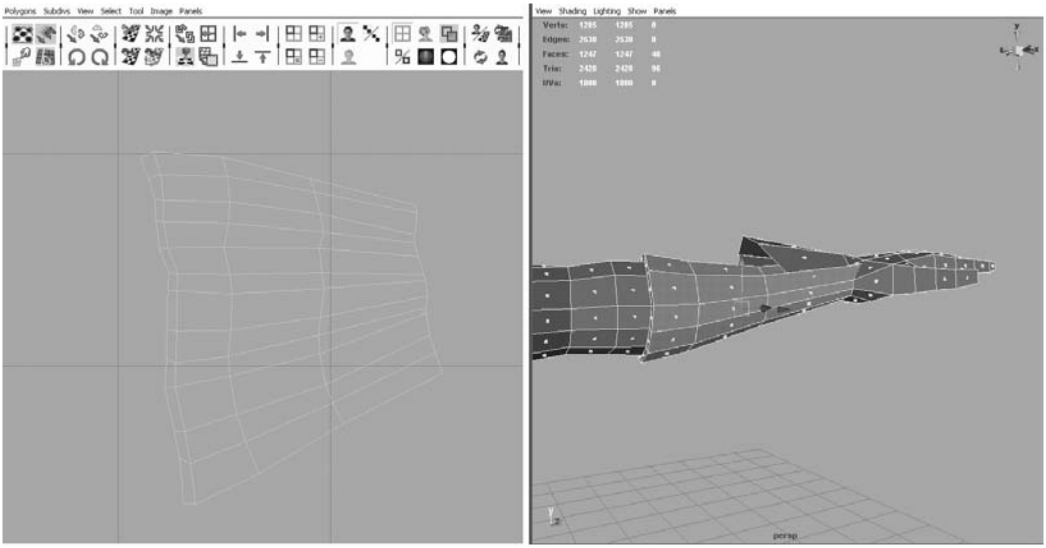


Figure 6.20
Work on the bracers next.

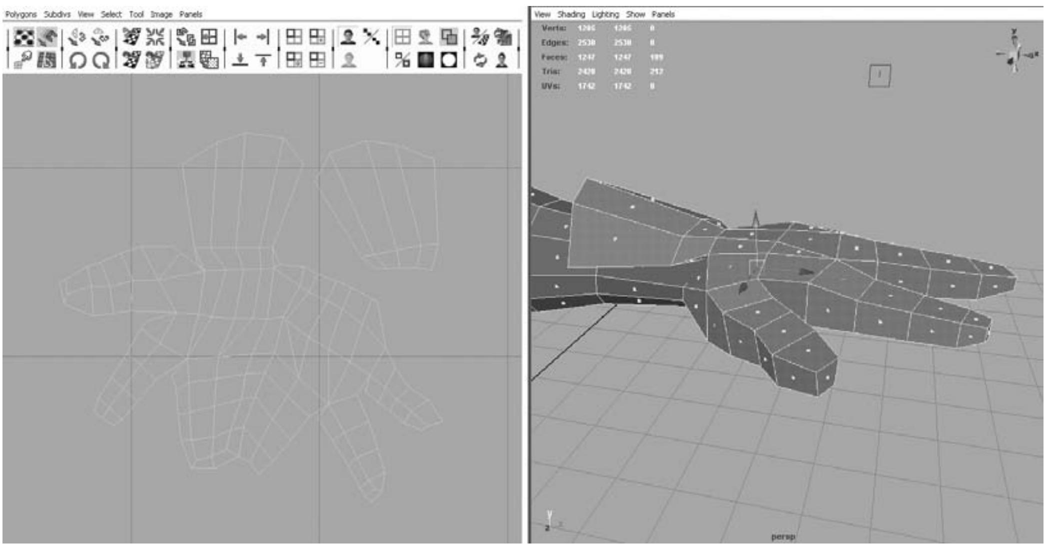


Figure 6.21
The UVs of the gantlet are more complex than other sections of the model.

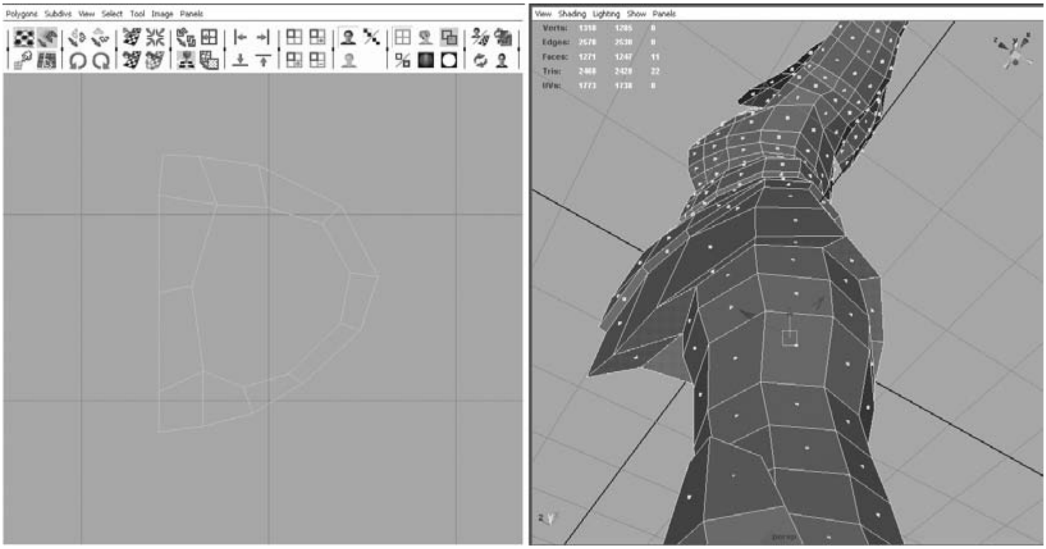


Figure 6.22
Don't forget to do the UVs under the tunic.

22. Put together the UVs of the chainmail portion of the character's leg, as shown in Figure 6.23.
23. Work on the lower leg plate, or *greave,* as shown in Figure 6.24.
24. The boots are a little more complicated; getting them to fit together will take some moving of the UVs. For guidance, look at how I organized the UVs for the boot in Figure 6.25.
25. You are almost finished. The remaining UVs should all be associated with the mouth. Look at Figure 6.26 for how I arranged the UVs of the mouth and tongue.

Organizing Texture Groups

You now have all the UVs organized into individual groups. The next step is to arrange the UV groups into separate texture groups. For this character, I will use four different materials, each with its own texture. The reason that I split the texture into four different groups is because each group has slightly different rendering attributes. For example, the flesh tones are not as shiny as the breast plate, but not as dull as the leather belt.

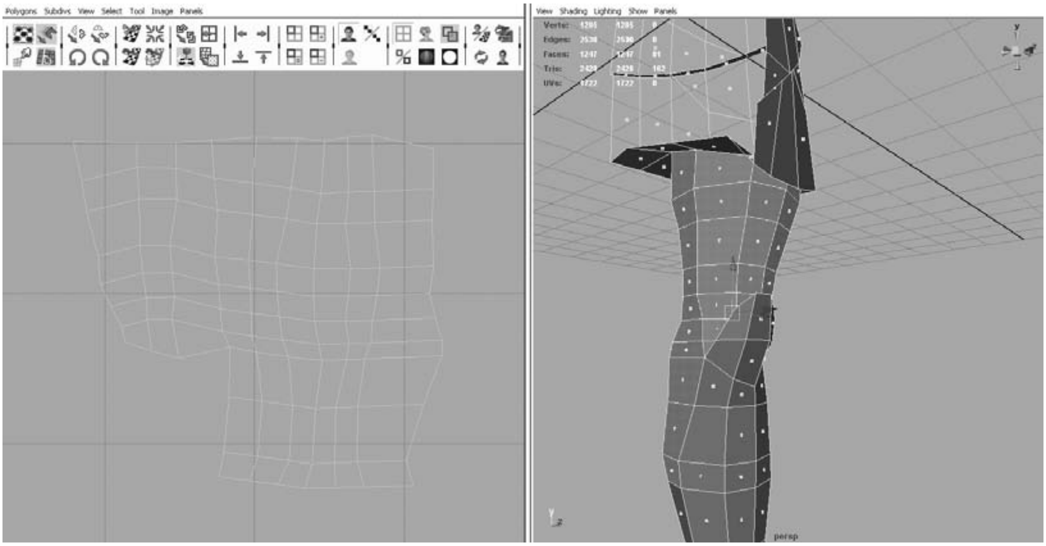


Figure 6.23
Organize the UVs associated with the chainmail portion of the character's leg.

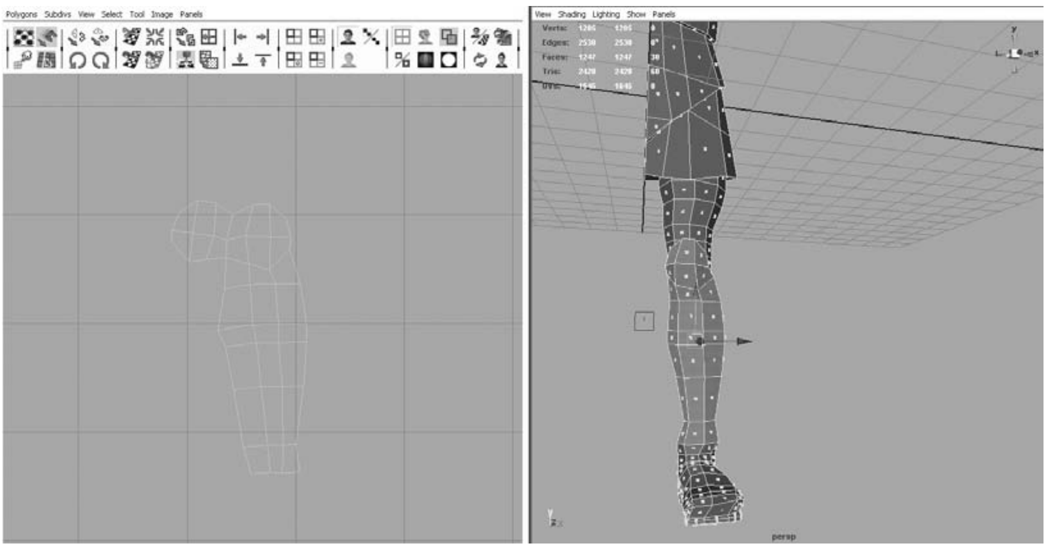


Figure 6.24
Organize the UVs of the greave.

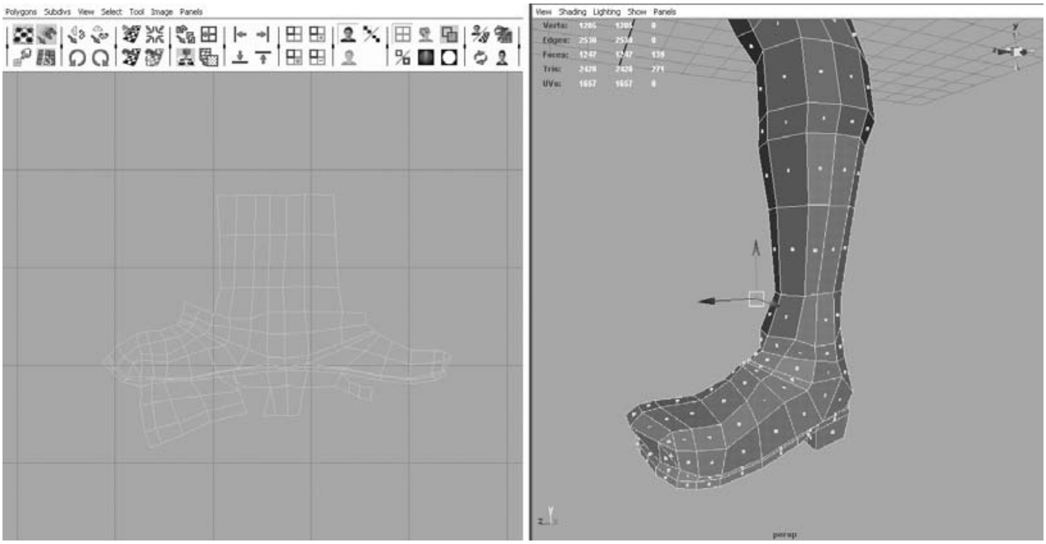


Figure 6.25
The boot will take some adjusting of the UVs to work.

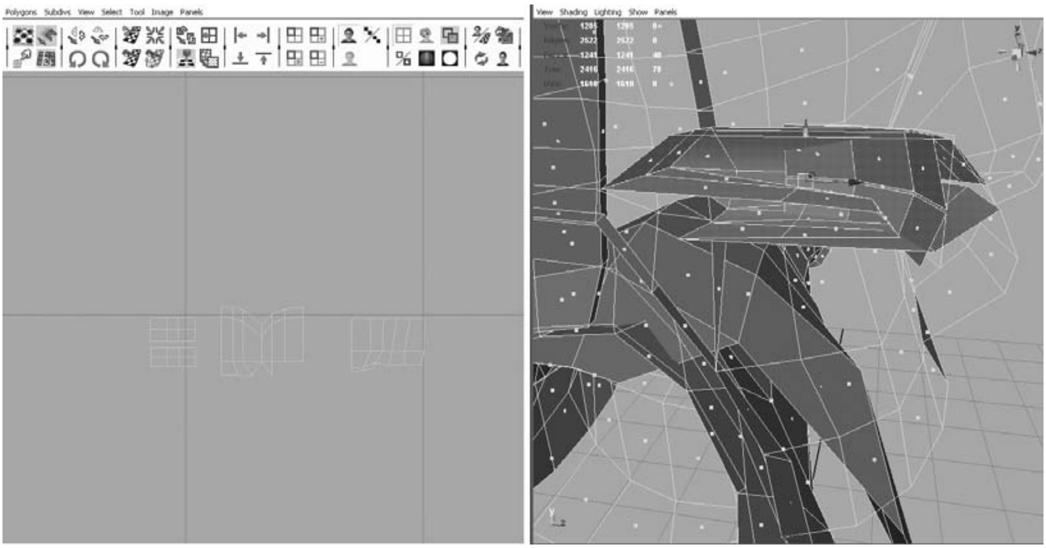


Figure 6.26
Set up the mouth as several different pieces.

1. Move the UVs of the face and the mouth to the 0 to 1 texture area. (This is the area on the upper-right portion of the grid that is a darker gray than the rest of the texture areas.)

2. Scale the UV groups to fit the texture area, as shown in Figure 6.27.

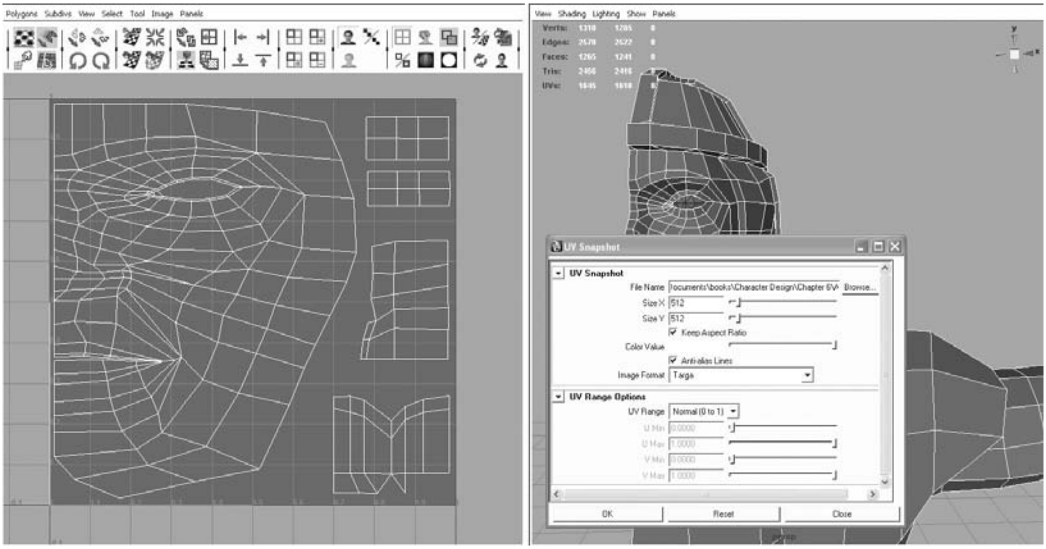


Figure 6.27
Arrange the UVs of the face in the texture area.

3. Notice on the right side of the screen shown in Figure 6.27 that I have opened the UV Snapshot dialog box; UV Snapshot takes a picture of the 0 to 1 texture area and saves it as a 2D bitmap image. To open this dialog box, use the Polygons menu in the UV Editor.

4. Specify where you want to save your 2D file and what you want to name it in the File Name field.

5. Change the Size X and Size Y settings to 512.

6. Make sure the Keep Aspect Ratio checkbox is checked.

7. Change the Color Value field to white.

8. Make sure the Anti-alias Lines checkbox is checked and that the Image Format field is set to Targa.

9. The UV Range field should be set to Normal (0 to 1).

10. Click the OK button to save the snapshot. You can see the results by opening the saved image in a 2D paint program such as Corel Painter, as shown in Figure 6.28.

11. Move the Face group down to the lower-right part of the grid, just below the 0 to 1 area.

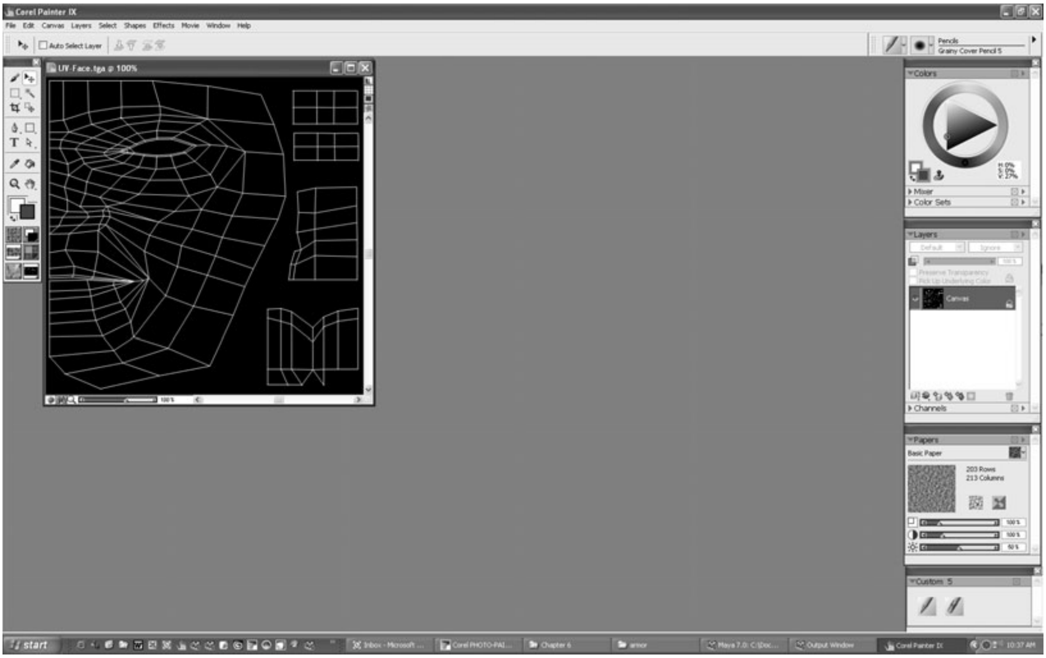


Figure 6.28
UV Snapshot creates a 2D image from the UV editor.

Note

You will be taking a snapshot of each of the four UV sets separately so that you can create separate texture maps for each. Each map then can have different attributes for surface qualities. Because the UV snapshot feature only take a snapshot of the 0 to 1 area you will have to move the UV sets into that area separately.

12. Create another texture group in the 0 to 1 area by moving all the plate armor pieces there, as shown in Figure 6.29. Take a snapshot of the UVs and then move them to the lower-left portion of the grid.
13. All the leather and fabric portions of the model go together as shown in Figure 6.30. Take a snapshot of these groups and then move them to the upper-left part of the grid.
14. The last section is for the chainmail. You'll put this section together a little differently because of the pattern nature of chainmail. To begin, move all the chainmail groups to the 0 to 1 area.
15. Snap the UVs to the grid, opening them up as shown in Figure 6.31. The idea is to have the chainmail texture repeat evenly across the UVs.

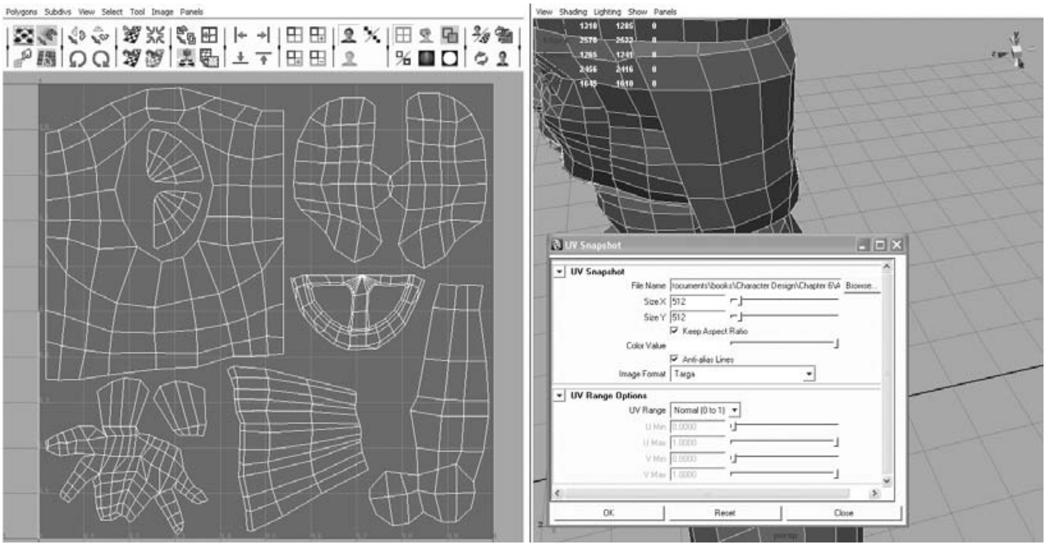


Figure 6.29
Arrange the plate metal groups into the texture area.

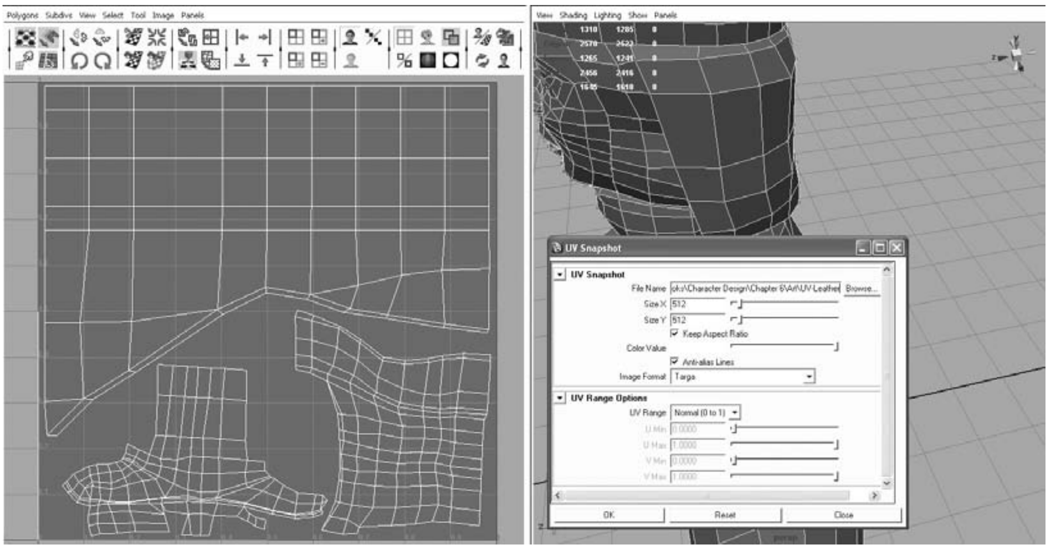


Figure 6.30
The leather and fabric portions all go together.

16. When you get to the chainmail at the back of the helmet, snap the inside portion directly to the outer portion as shown in Figure 6.32. (Snap to Grid works in the same way in the UV Editor as it does in modeling. Just hold the X key on the keyboard down while moving the UV.)

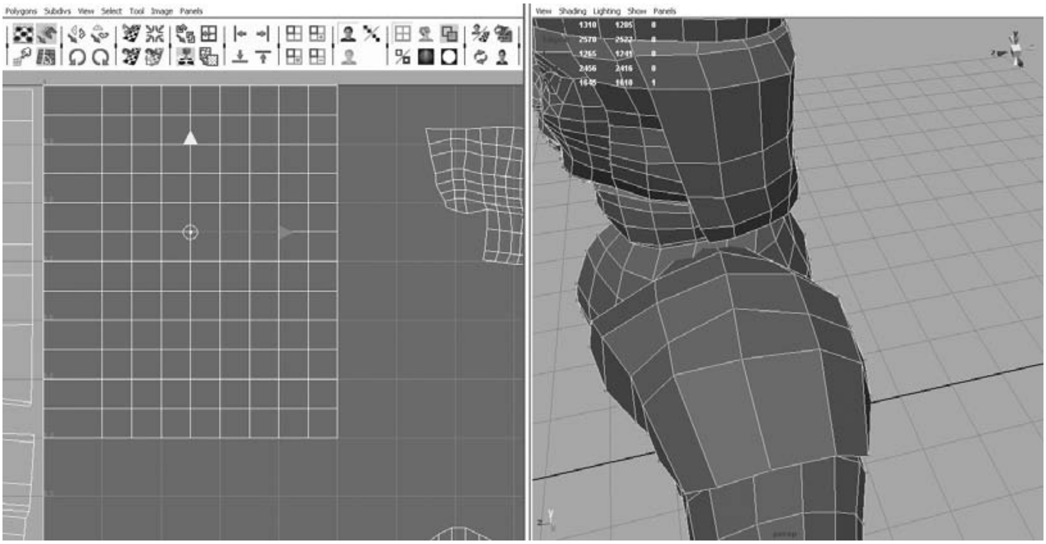


Figure 6.31
Snap the UVs to grid coordinates.

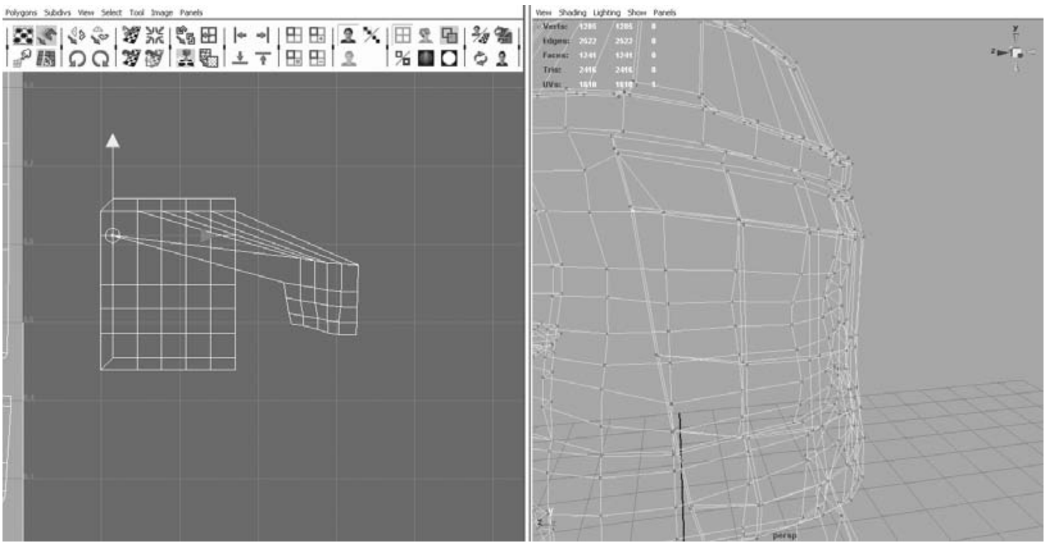


Figure 6.32
Snap the bottom and top of the helmet chainmail together.

17. When you are finished, your UV map of the chainmail portions of the model should look similar to Figure 6.33.

18. After you have the UVs spread out in a grid, you can scale them as needed. You should now have four texture sets that look similar to those shown in Figure 6.34.

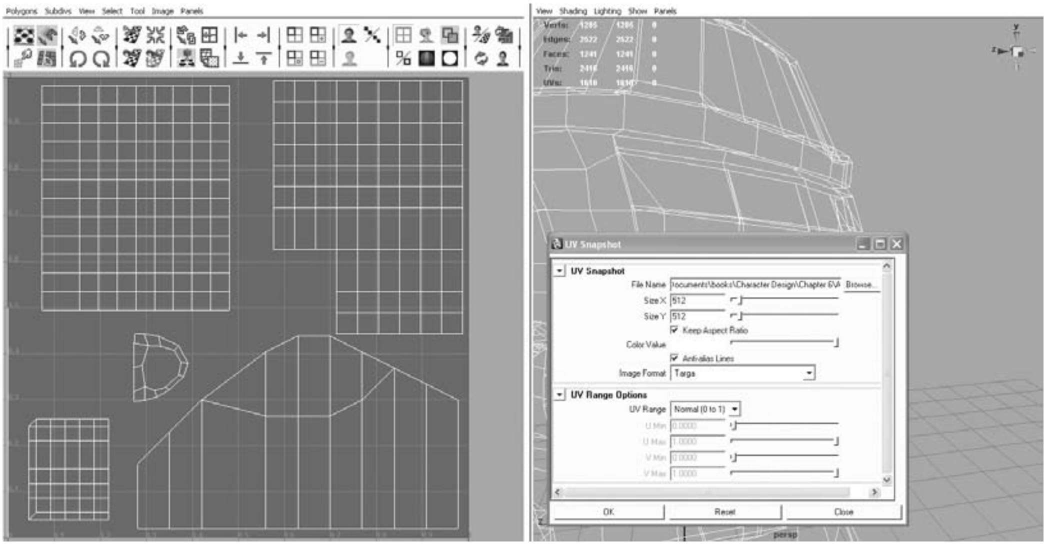


Figure 6.33
Arrange the UVs for the chainmail in the 0 to 1 area.

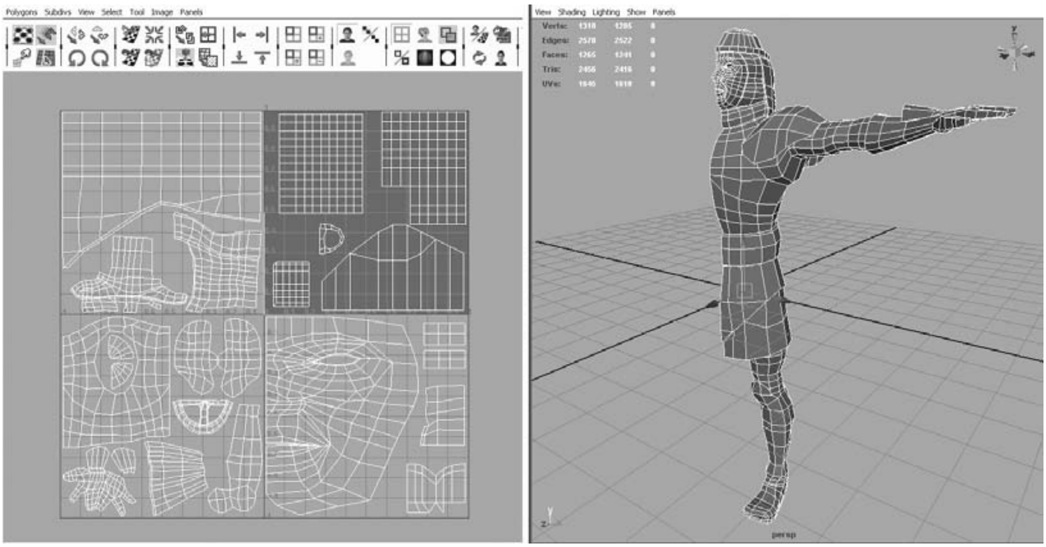


Figure 6.34
The leather and fabric portions all go together.

Note

For optimum texture rendering, Maya suggests that you always keep your UVs in the 0 to 1 texture area (the darker grey area in the UV Editor). Some game engines may require the UVs to be in this area, but not all will. Having them split into four separate groups helps in the UV selection process. If you need to move them into the 0 to 1 area, wait until the end of the texturing process.

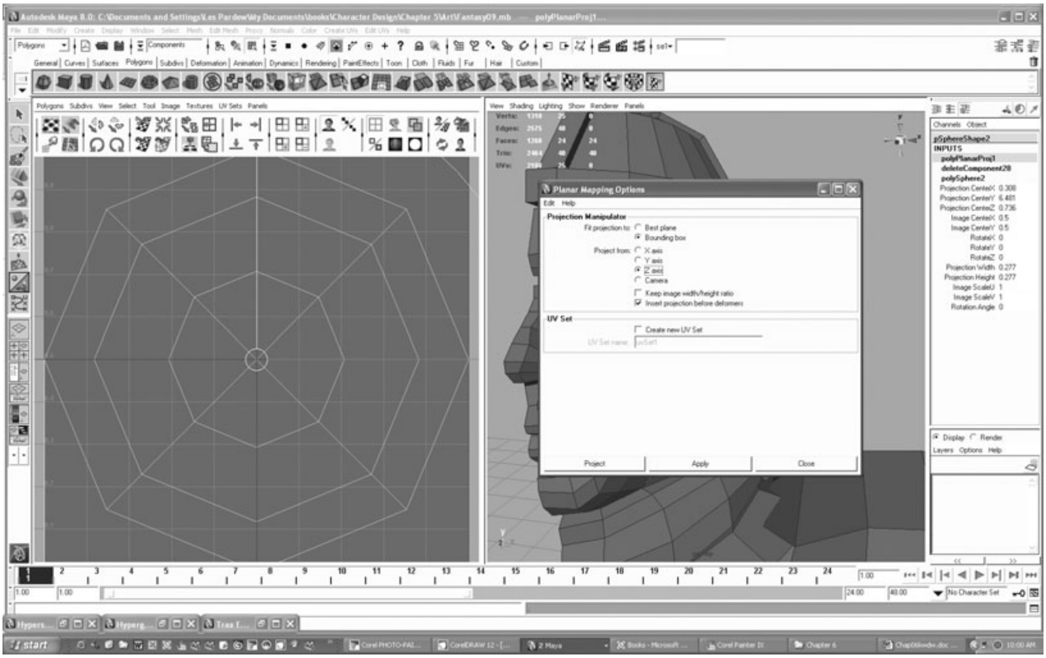


Figure 6.35
Use the Planar Mapping tool to map the UVs of the eyeball.

19. There is one last part of the model that still needs to have its UVs organized: the eyeball because it is a separate object. For this part of the model, you can use simple planar mapping, as shown in Figure 6.35. The Planar Mapping tool is located in the Create UVs directory.

Painting the Textures

After all the technical work you have just completed, you finally get to move on to something that is more artistic. Painting the textures of a game character is probably the closest thing there is in the character-creation process to traditional art. In essence, you are painting the surface of your model, but because the UVs are mapped to a flat surface, you can create the textures as if you were painting a picture.

Note

In this section, I use Corel Painter to paint the textures. Corel Painter is very similar to Photoshop, but has a number of features that simulate natural media. Most of what I show here in Painter can also be accomplished in Photoshop, so you should be okay if you don't have Painter. Paint Shop Pro is also a good painting program.

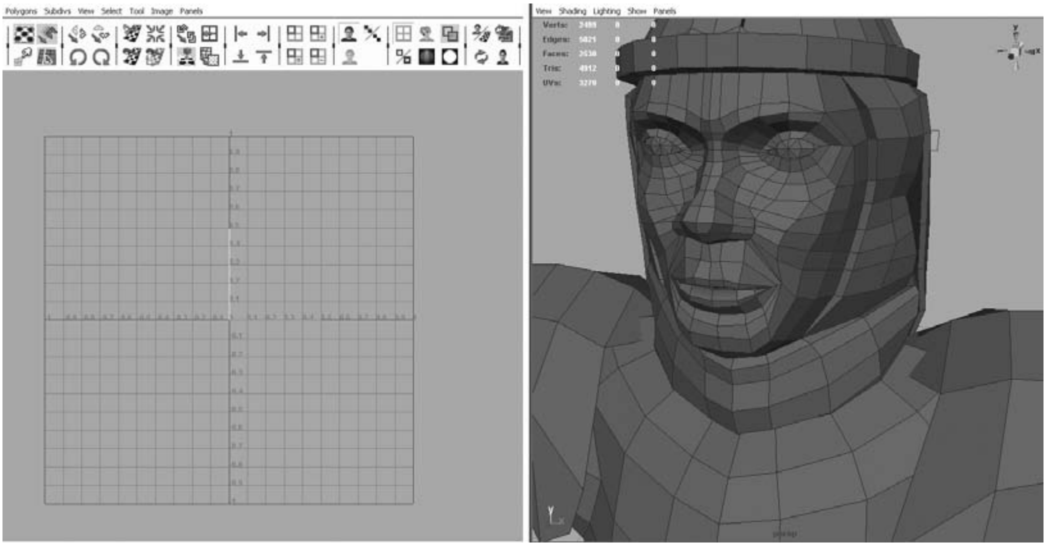


Figure 6.36
Mirror the model to complete the other half.

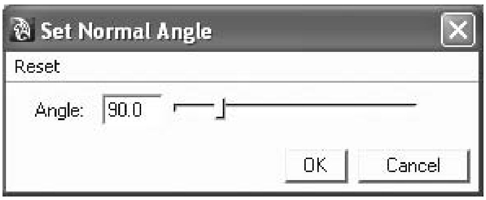


Figure 6.37
Change the Angle setting to 90.

Preparing the Normals

Now that the UVs are organized, you can mirror the model and merge the edge vertices like you did in the last chapter. The new UVs for the mirrored half of the model will be placed exactly over the current UVs. Figure 6.36 shows the mirrored model.

If you haven't noticed already, the model looks very faceted. That's because it is made up of a lot of flat polygons. You will need to soften the edges if you want your character to look smooth. To begin, select the model open the Edit Mesh menu, choose Set Normal Angle. Change the Angle setting to 90 as shown in Figure 6.37 and click OK.

90 is a good setting for any model that needs to look round. Unfortunately, rounder surfaces can be a problem for those polygon edges that need a sharp or

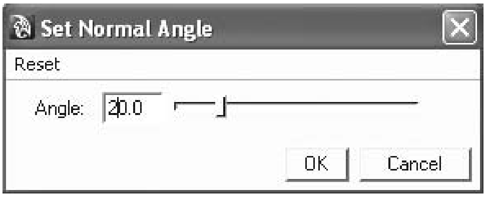


Figure 6.38
Select the edges that need to be hard and change their Angle setting to 20.

hard edge, like those of the helmet. For these types of areas, select the edges that should be hard (here, the ones around the brim of the helmet), launch the Set Normal Angle options dialog box, and change the Angle setting to 20, as shown in Figure 6.38.

You will need to go through the entire model and change the edges that should be hard by hand. Yes, I know it's a pain, but there's no better way to do it.

Painting the Face Texture

Whenever I am drawing or painting, I always start with the face; I begin there when painting textures as well. Of course, if you were creating a model strictly for rendering, you could add a lot of detail to the face. Because this is a video game, however, you are limited to a small texture space. Complicating this is the fact that human flesh tones are among the most difficult tones to paint because human skin is somewhat transparent. That is, light passes through the upper layers of the skin before it bounces back. This passing through the upper layers of the skin causes the light to scatter or refract in different directions, and gives the skin a translucent quality.

1. You will create multiple layers for the texture of the skin, with the bottom layer being the UV snapshot you took in the preceding section. Load that file into Painter.
2. Next, you need to create a skin-tone layer. Create a new image in Painter by opening the File menu and choosing New.
3. Adjust the color of the new image to a flesh tone, as shown in Figure 6.39.
4. Copy and paste the new flesh-colored image over the UV snapshot.
5. Set the Transparency slider found at the top of the Layers palette to .96 so that you can see the UVs through the new layer, as shown in Figure 6.40.

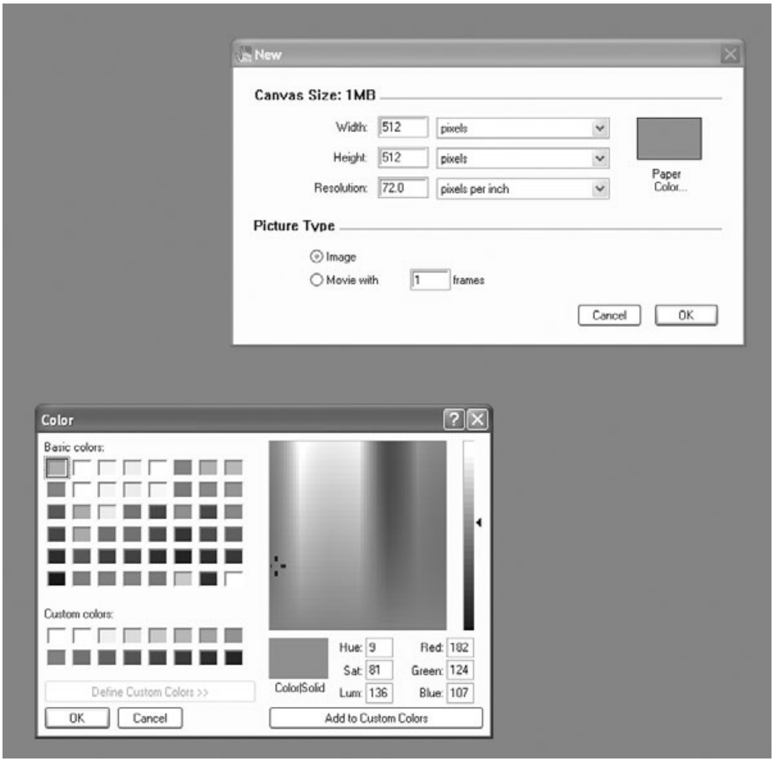


Figure 6.39
Change the color of the new image to a neutral flesh tone.

6. With the Airbrush tool, lightly paint in the various base colors of the face, as shown in Figure 6.41.

Note

When painting a texture, you need to know the power of the game engine with regard to lighting. Most game engines support real-time lighting, but some of the older ones do not. If your game engine supports real-time lighting, you need not paint a lot of light and shadow in your texture because the engine will handle that. If, on the other hand, your game engine does not support real-time lighting, you will need to simulate lighting in your textures by painting in the light and shadow areas as if you were painting a portrait.

7. Paint in the colors for the face in more detail, as shown in Figure 6.42.
8. Paint in the mouth, as shown in Figure 6.43.
9. Add a little surface variation to give the tongue a rougher appearance. To begin, duplicate the flesh-tone layer by opening the Layers menu and choosing Duplicate Layer.

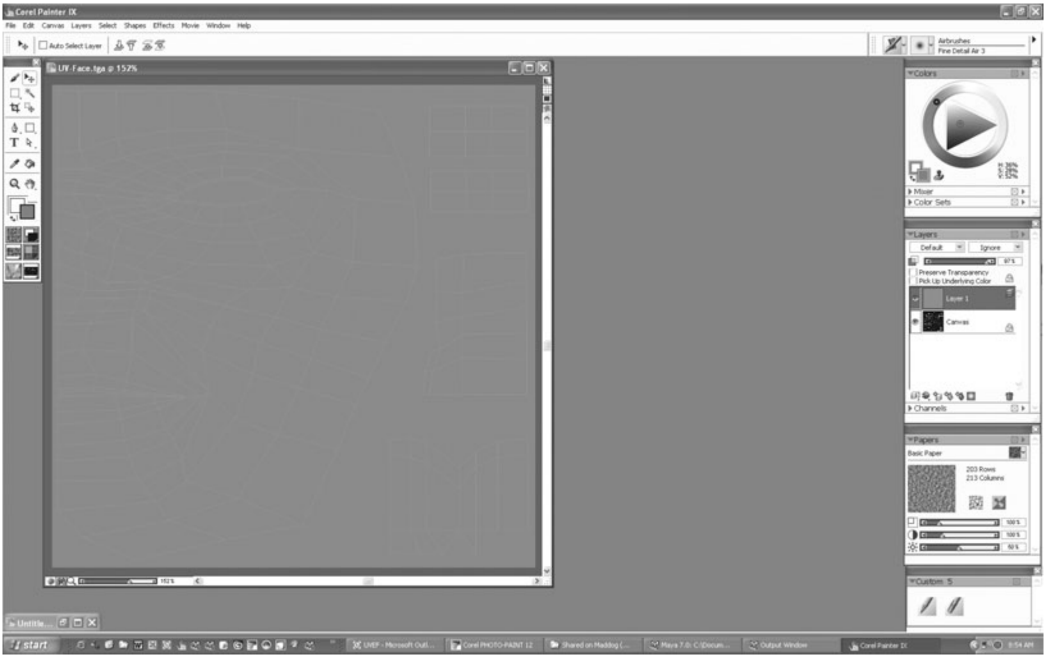


Figure 6.40
Set the new layer's transparency to .96.

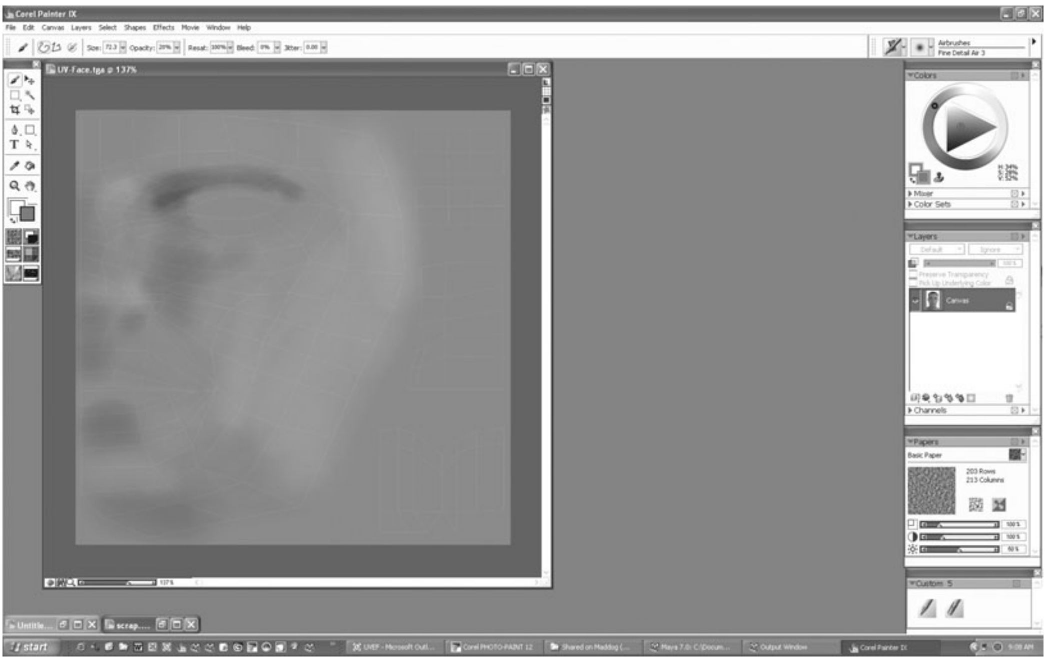


Figure 6.41
Paint in the base colors of the face.

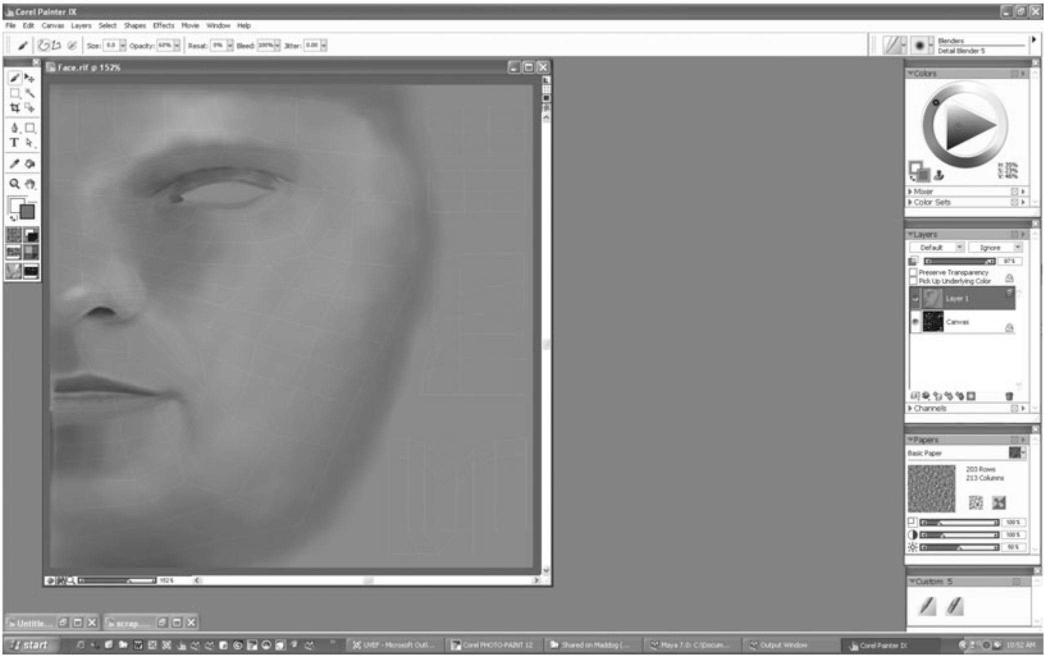


Figure 6.42
Add more detail to the face.

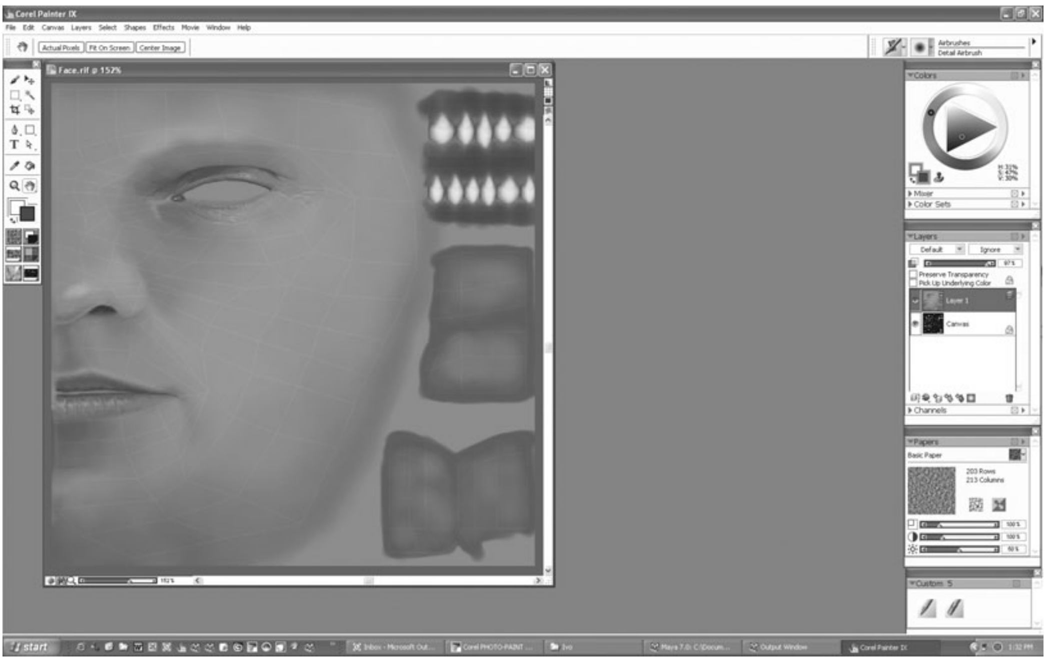


Figure 6.43
Paint the teeth and mouth.

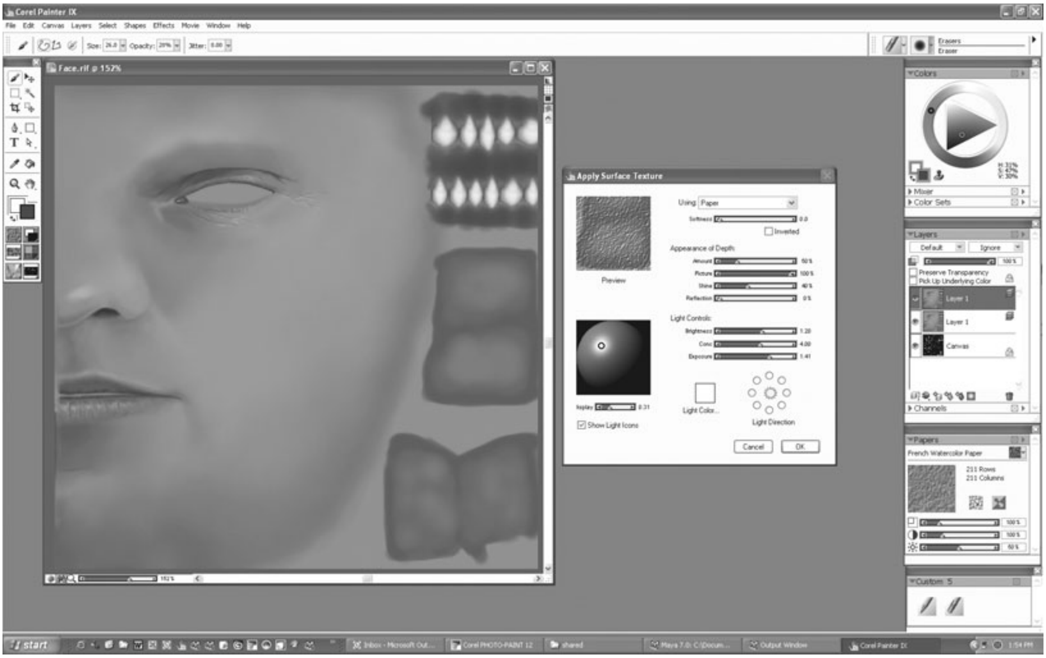


Figure 6.44
Adjust the surface texture for the tongue.

10. Select new layer by clicking it in the Layers palette.
11. Open the Effects menu, choose Surface Control, and select Apply Surface Texture to open the Apply Surface Texture dialog box. You'll see the effects of your surface texture settings in the preview box.
12. Select French Watercolor Paper either from the toolbox or from the Papers palette and adjust the sliders to give the tongue a rough, shiny texture, as shown in Figure 6.44.
13. Use the Eraser brush to erase the entire new layer except the tongue, as shown in Figure 6.45.
14. Duplicate the flesh-tone layer and create on it a texture for the mouth using the Basic paper, as shown in Figure 6.46.
15. Erase all the areas except the mouth.
16. The character needs just an indication of stubble where his beard is. Again, duplicate the flesh-tone layer and apply a surface texture. This time, use the

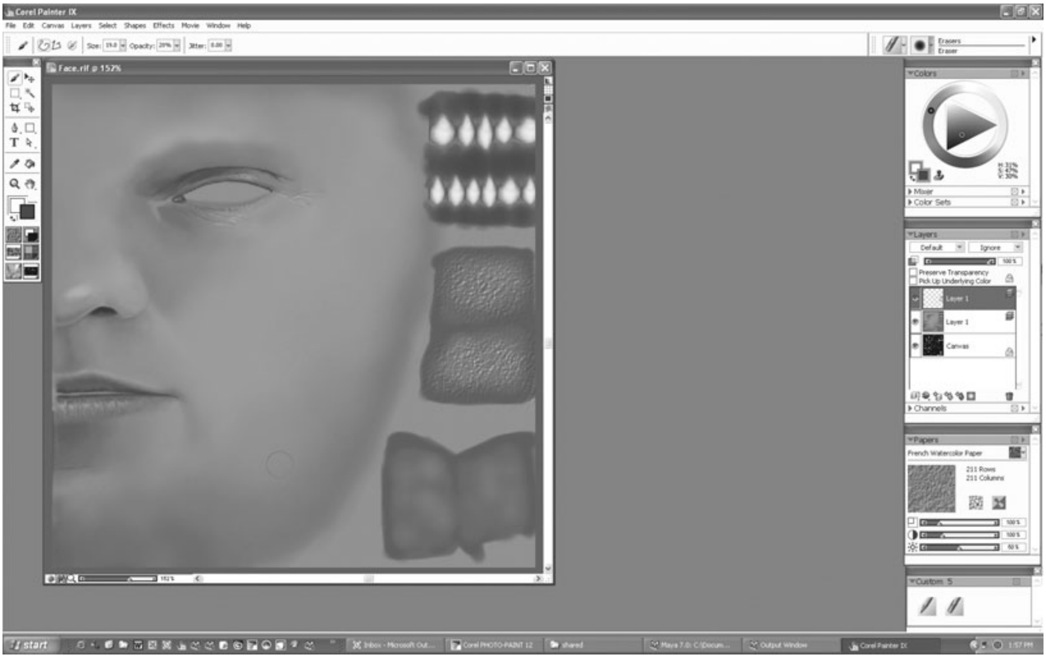


Figure 6.45
Erase all the areas that do not need the tongue texture.

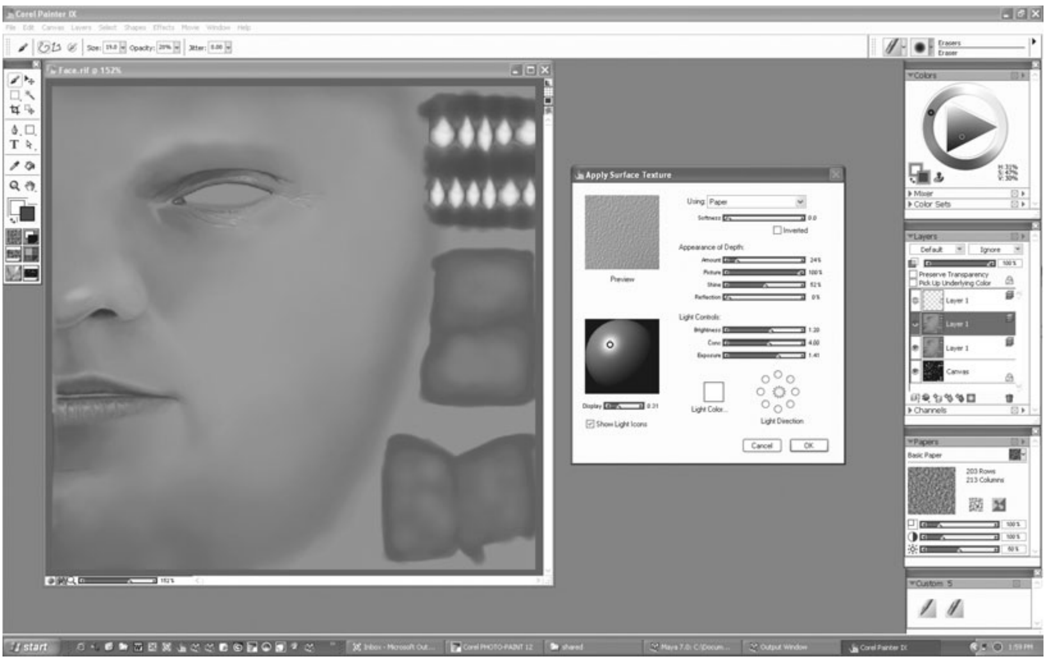


Figure 6.46
Use the Basic paper to create a texture for the mouth.

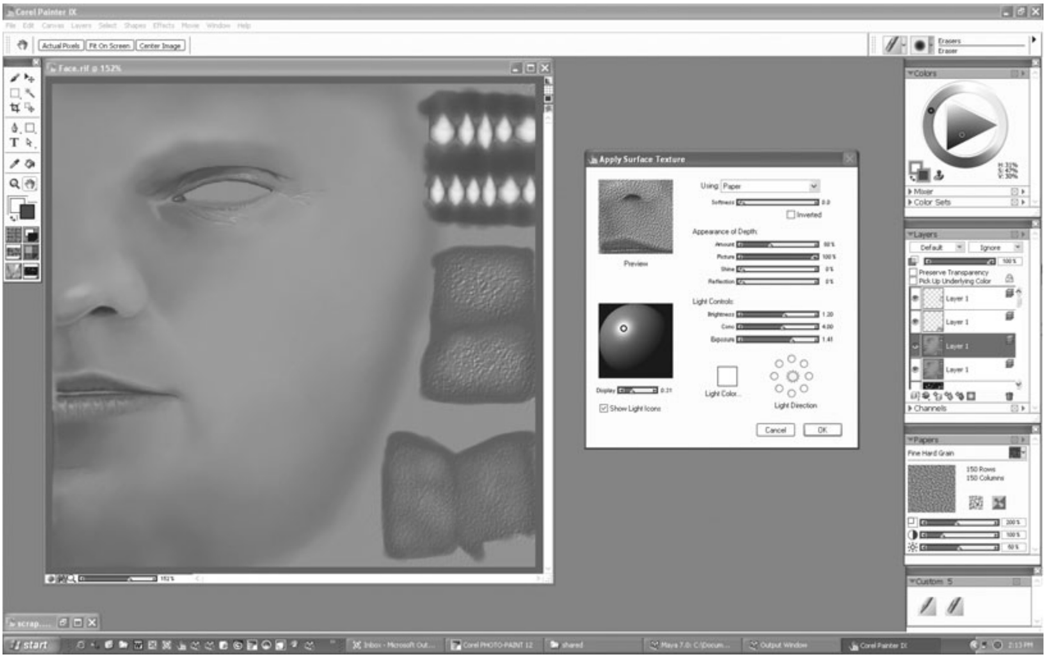


Figure 6.47
Create a stubble texture for the character's beard.

Fine Hard Grain paper. Adjust the Amount slider to the right and Shine slider to the left to create a coarse texture, as shown in Figure 6.47.

17. Erase all areas except along the jaw and mouth where the character's beard would grow, as shown in Figure 6.48.
18. Duplicate the flesh-tone layer again and place it above the stubble layer by moving it up in the stack on the Layers palette, as shown in Figure 6.49.
19. Give the new layer a more subdued texture for the pores in the skin using the same paper you used for the stubble but reducing the Amount setting, as shown in Figure 6.50.
20. Use the Erase tool to remove the texture from those areas where it is not needed, blending where it joins with the texture of the beard (see Figure 6.51).
21. The beard area needs to be a much cooler color than the rest of the face. Create a new layer at the top of the layer stack by opening the Layers menu and choosing New Layers.

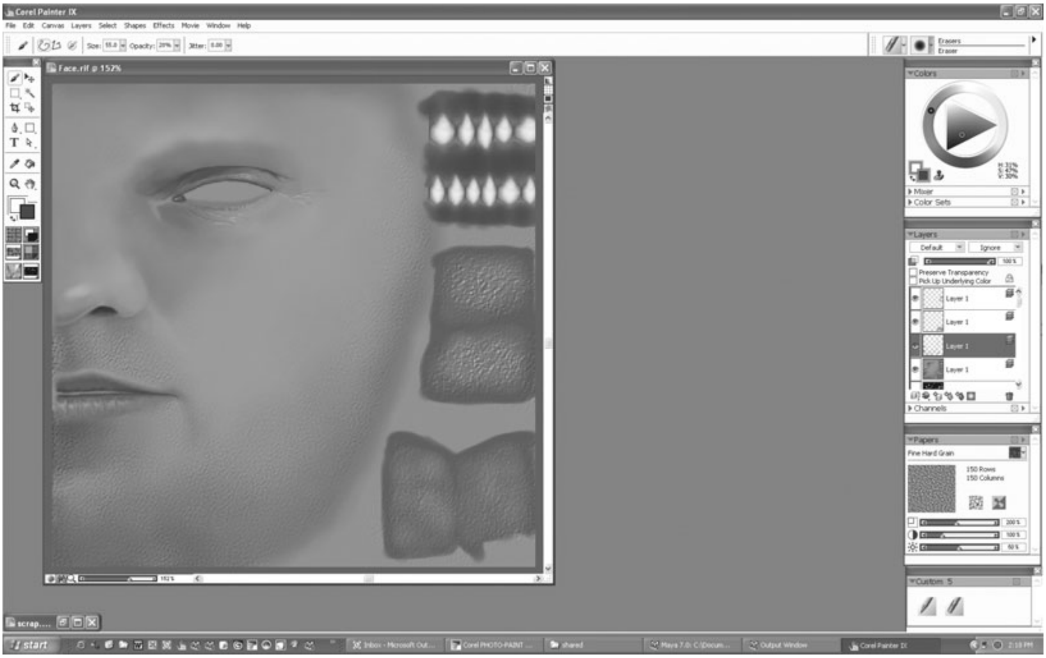


Figure 6.48
Erase the texture around the beard area.

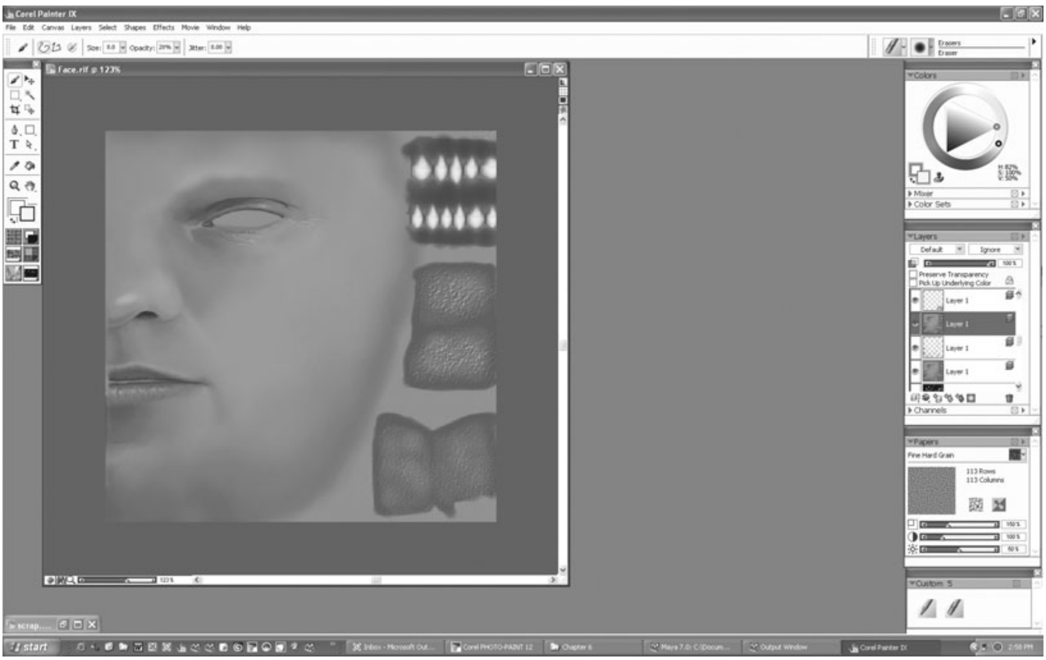


Figure 6.49
Move the new layer up in the stack.

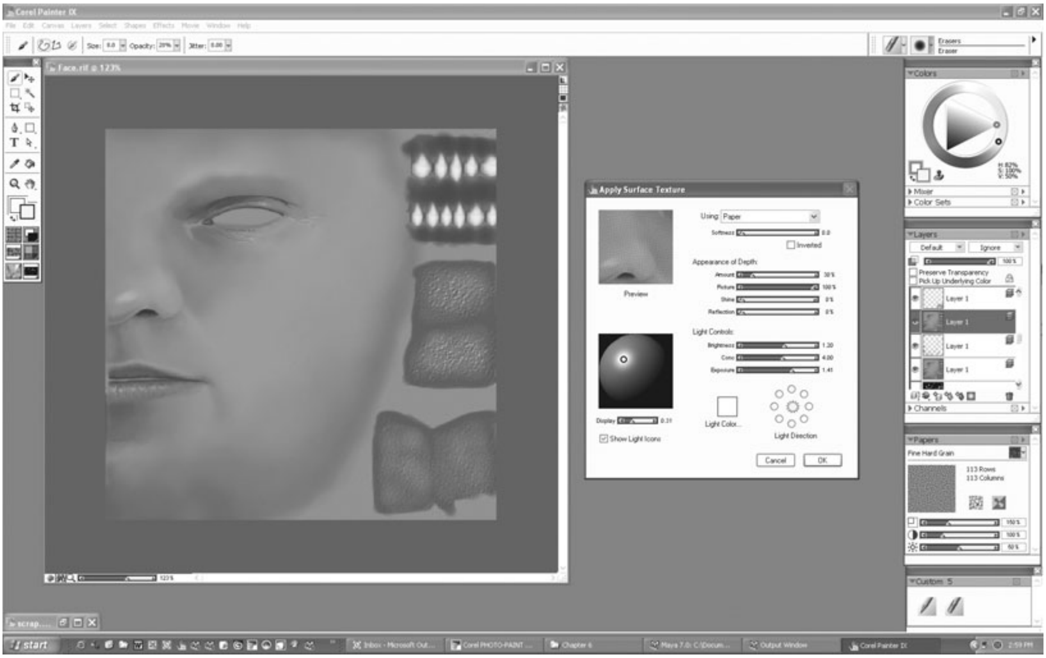


Figure 6.50
Add pores to the skin.

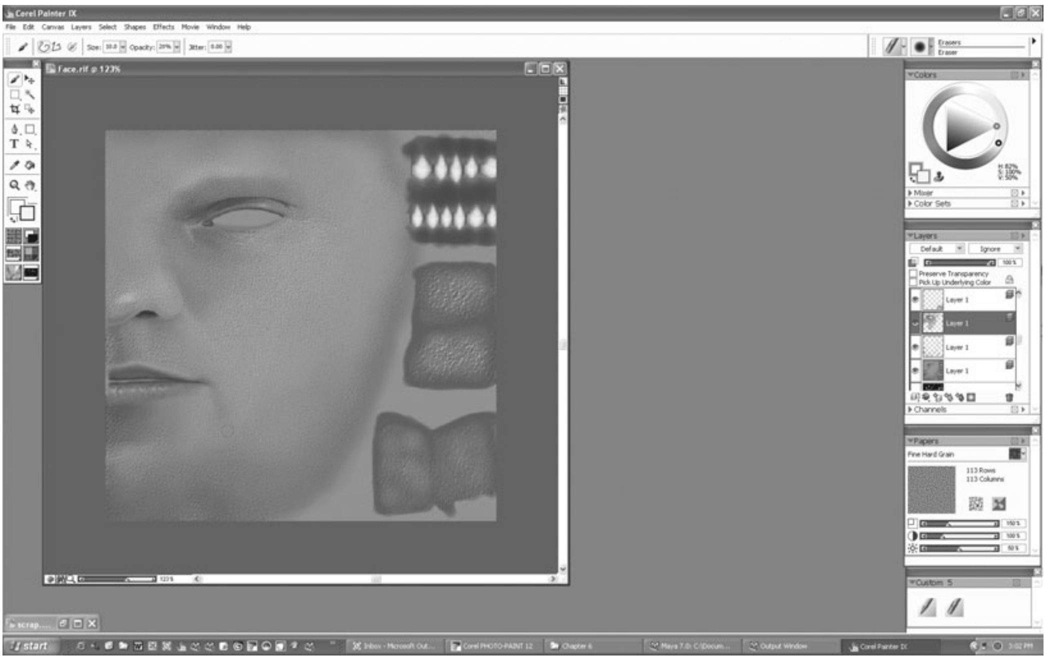


Figure 6.51
Apply the new texture only to those areas of the face that need it.

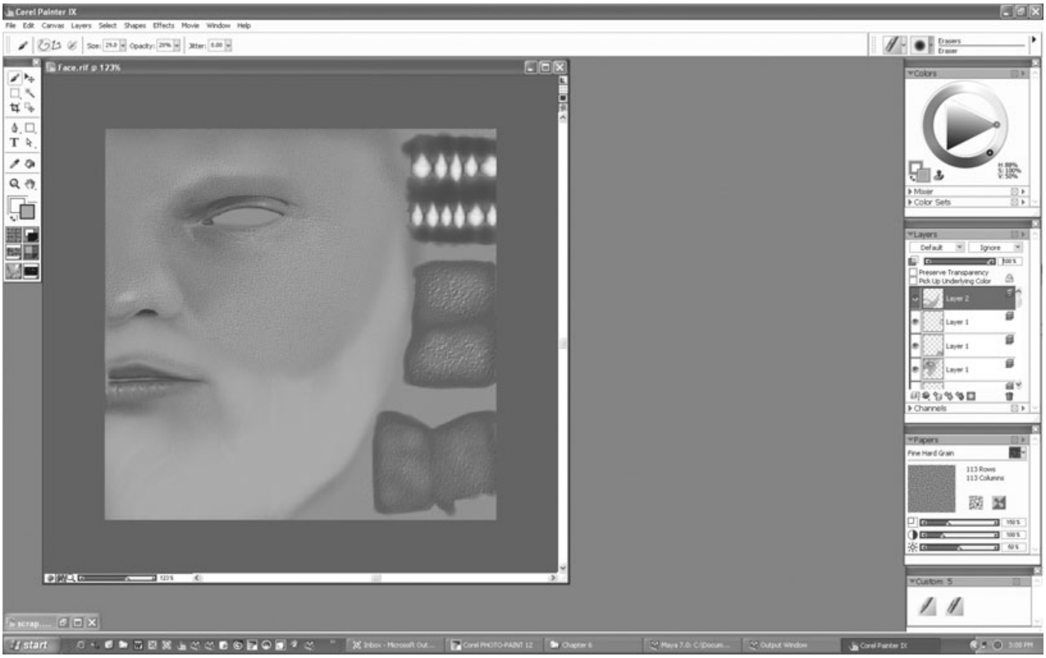


Figure 6.52
Paint the beard area blue.

22. Paint the beard area blue, as shown in Figure 6.52.
23. Change the blue layer's Transparency setting to .5, as shown in Figure 6.53.
24. Open the Layers menu and choose Drop All to flatten all the layers to one and save the file as a BMP file.

The face texture is now ready for testing; you'll need to load it into Maya to see how it looks on the model. (You should test all your textures this way.)

Loading a Texture into a Material

Now the model is ready to have a material applied to the character. Materials in Maya have a number of attributes. Depending on the game engine, some of the attributes may or may not be included when they are exported to the game. I will stick to the most common ones here.

1. First, create a new material, Hypershade. Hypershade is located under Rendering editor in the Windows directory.
2. In the Hypershade window, open the Create menu, choose Materials, and select Blinn to create a new Blinn material.

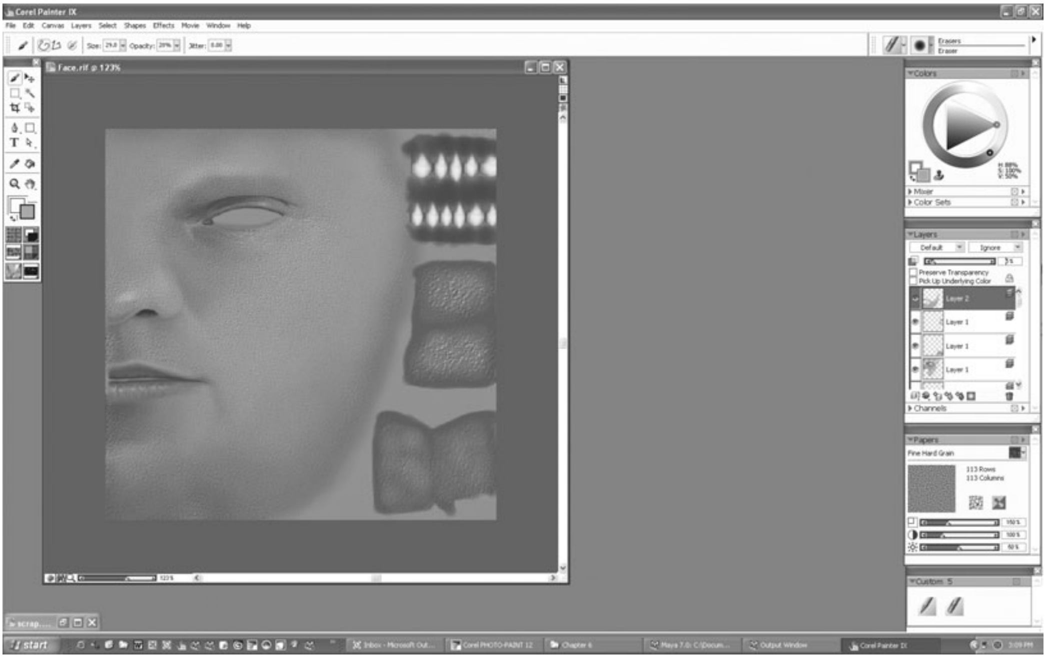


Figure 6.53
Make the blue layer very transparent.

3. Rename the new material Face and load the face texture the same way you loaded the textures for the templates when you started the model in the last chapter (see Figure 6.54).

4. An easy way to select the faces to which you want to apply the textures is to select them in the UV editor. (Note that I'm talking about the polygon faces here, which happen to be on the character's face.) Change the selection mode to Face and drag a selection box around the Face group, as shown in Figure 6.55.

5. With the polygon faces selected, open the marking menu over the material in the Hypershade window and choose Apply Material to Selected. The character's face material should appear on the model as shown in Figure 6.56.

6. Because a Blinn material is typically used for metal objects (indeed, you will use it later for your character's chainmail and plate armor), you must adjust several settings in order to use it for a skin texture. In particular, you must reduce the Eccentricity setting and eliminate the Specular Roll Off feature. You will also need to increase the Translucence setting to about .5, as shown in Figure 6.57.

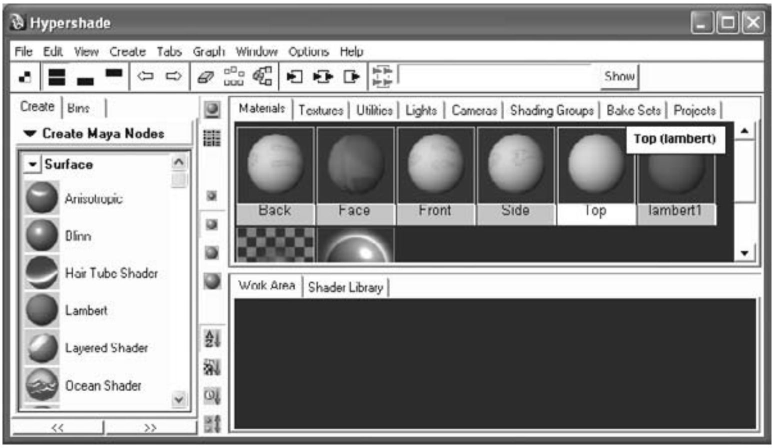


Figure 6.54
Load the face texture into a new Blinn material.

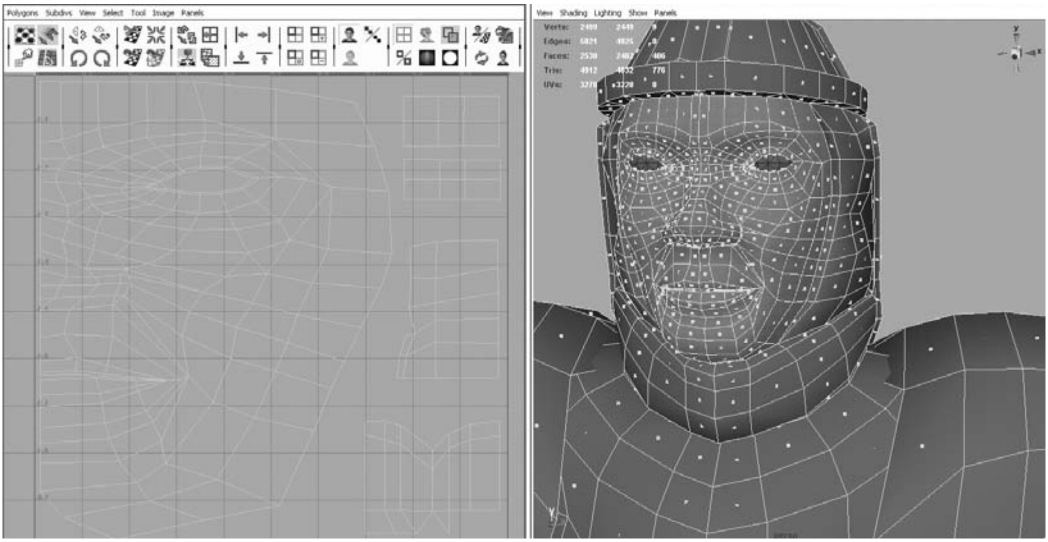


Figure 6.55
Select the Face group in the UV editor.

7. The real-time render in Maya doesn't do a very good job of showing the changes to the Blinn material. To see the changes, you need to do a test render of the model. Change the UV editor panel to the Render View panel and, under Render/Render, select Persp to render a view of the model using the default lights. You can adjust the setting to the material until you get a look that works for a skin texture.

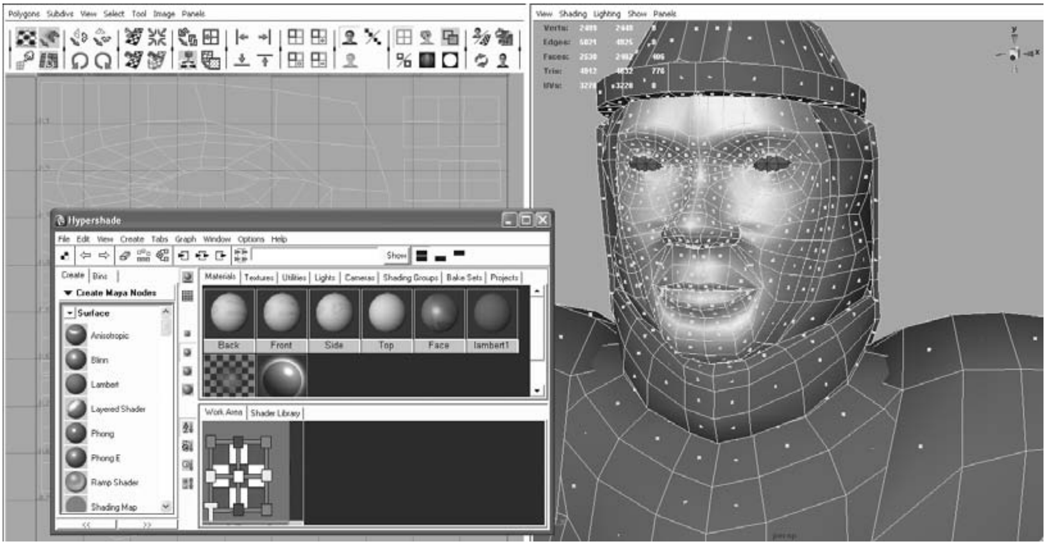


Figure 6.56
Apply the material to the selected faces.

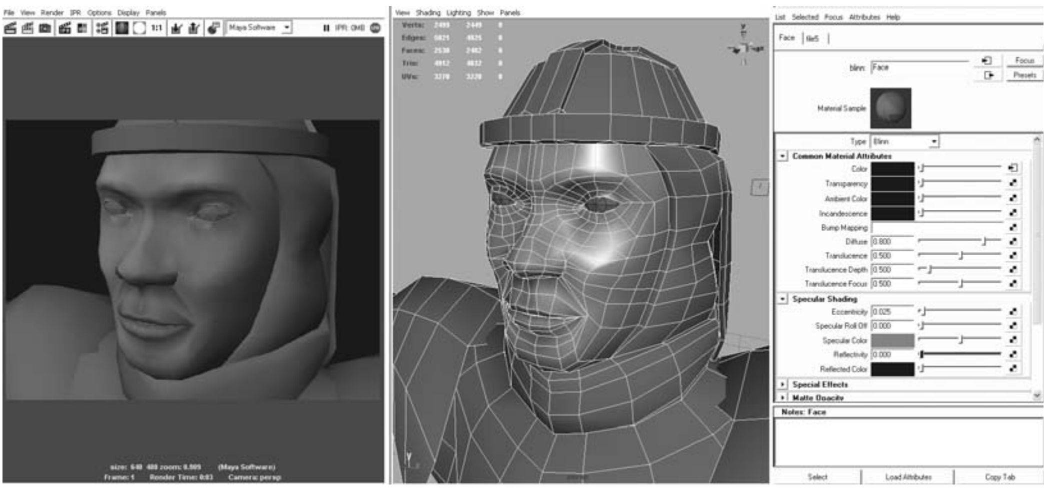


Figure 6.57
Change the setting of the Blinn material to work better for a skin texture.

8. Go back to the face texture in Painter and add the final detail of eyebrows and eyelashes, as shown in Figure 6.58. Save the texture.

9. Update the texture in Maya by clicking the Reload button (assuming the name of the file has not changed). This is a great way to work on your textures—just update them as you go (see Figure 6.59).

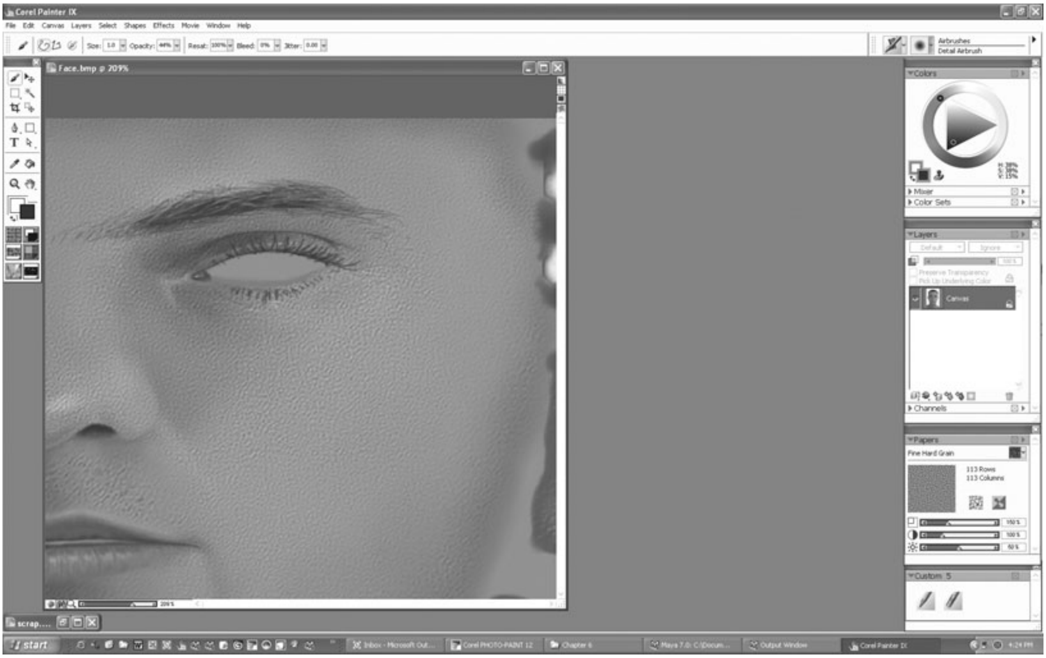


Figure 6.58
Paint in the eyebrows and eyelashes.

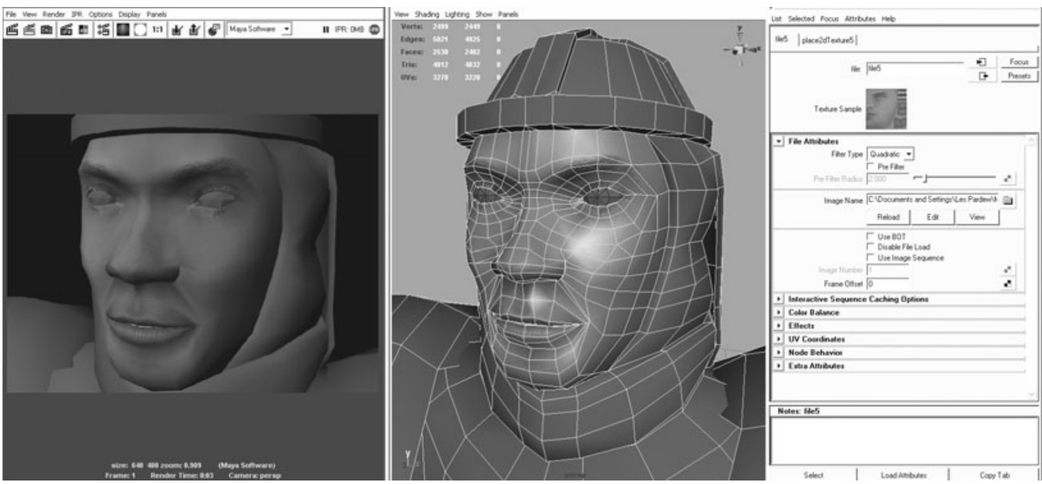


Figure 6.59
Update the texture by using the Reload button.

Painting the Chainmail Texture

The chainmail texture is a little different from the other textures you will create for the character because it is basically a pattern texture. You can see in Figure 6.60 the pattern I created for the chainmail texture.

To create your own chainmail pattern, do what I did: paint a single link section and then repeat it across the entire texture. Then do the following:

1. Create a new material and load the chainmail pattern.
2. Apply the material to the chainmail group to see how the chainmail looks. It should look similar to my model, shown in Figure 6.61.
3. You may need to adjust a few UVs to get the chainmail on the model to look just right. Figure 6.62 shows irregularities of the UVs of the leg.
4. You may also need to adjust the size of your UV groups to try to get them to all look consistent, as shown in Figure 6.63.

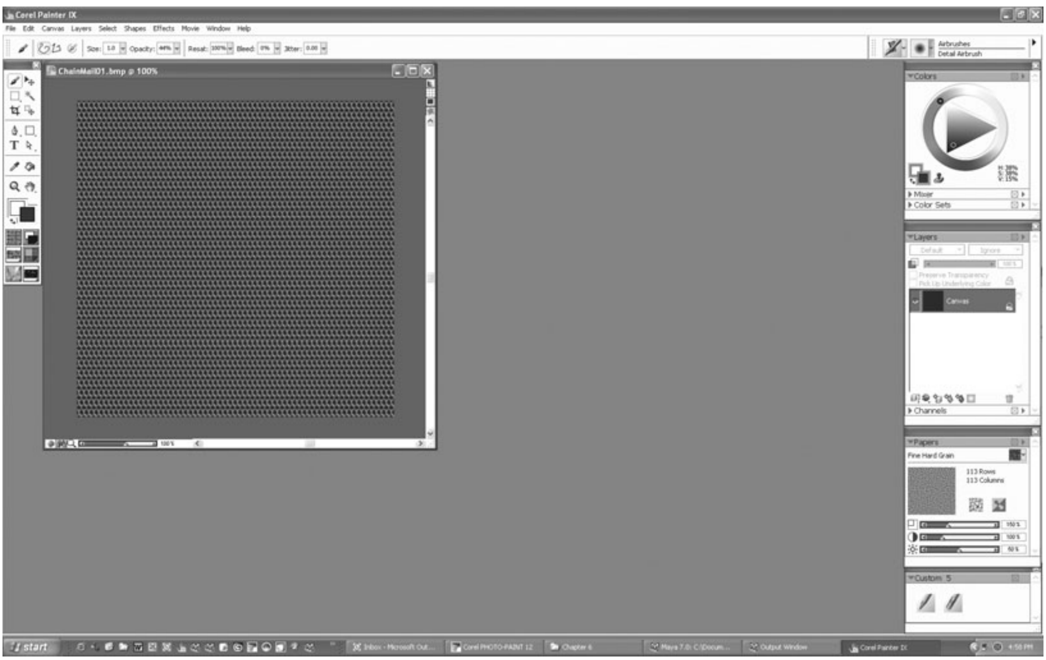


Figure 6.60
A chainmail pattern.

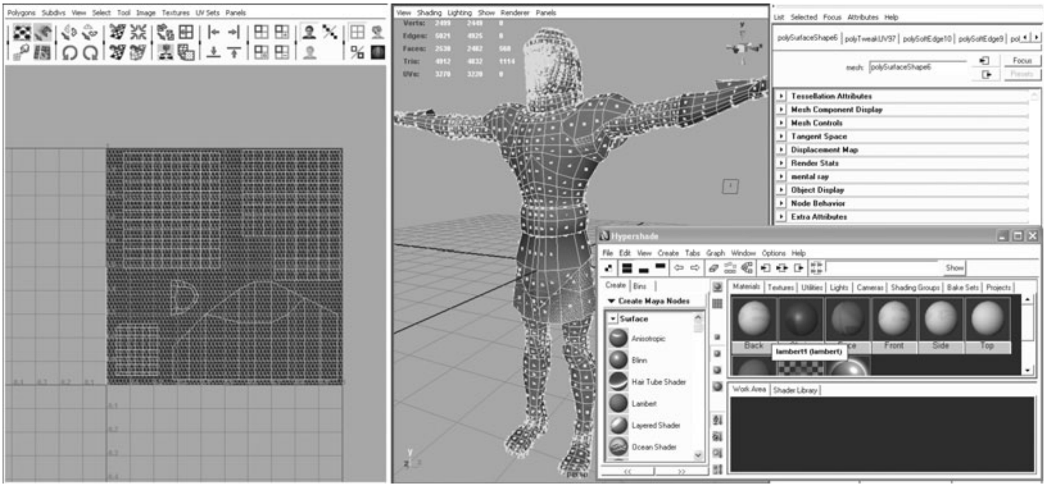


Figure 6.61
Apply the chainmail material to the chainmail group.

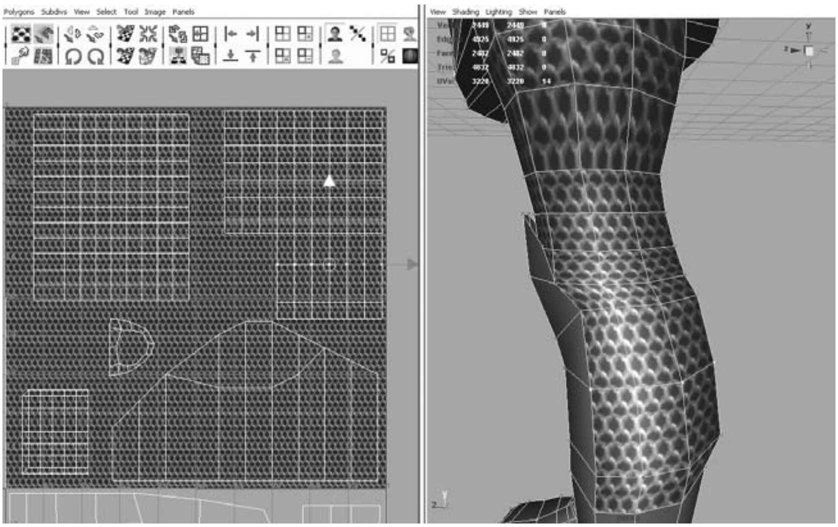


Figure 6.62
Adjust the UVs in the leg to give the chainmail a more even appearance.

5. When the chainmail looks right, go back and add a few shadows and a leather strap that holds the greaves to the character's leg. Figure 6.64 shows the additions. I got the shadow by duplicating the chainmail layer and then darkening the layer on the bottom. Then I just erased the shadow areas from the top layer.

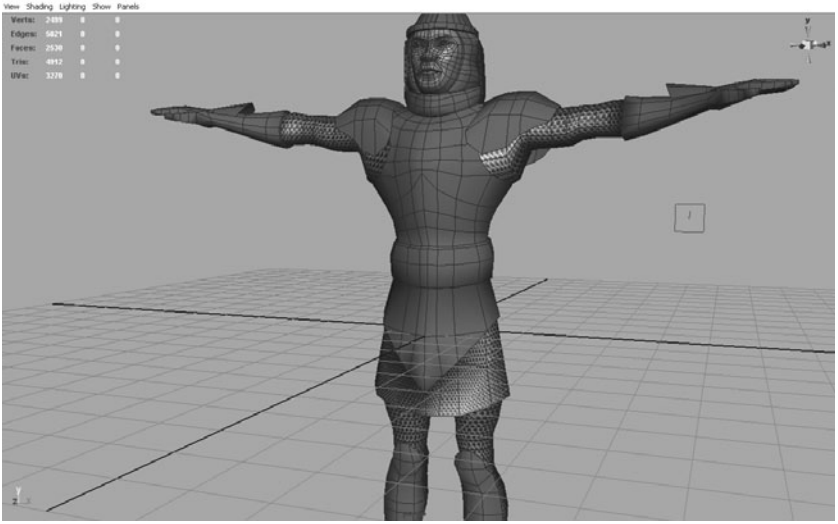


Figure 6.63
Adjust the scale of the UV groups to match.

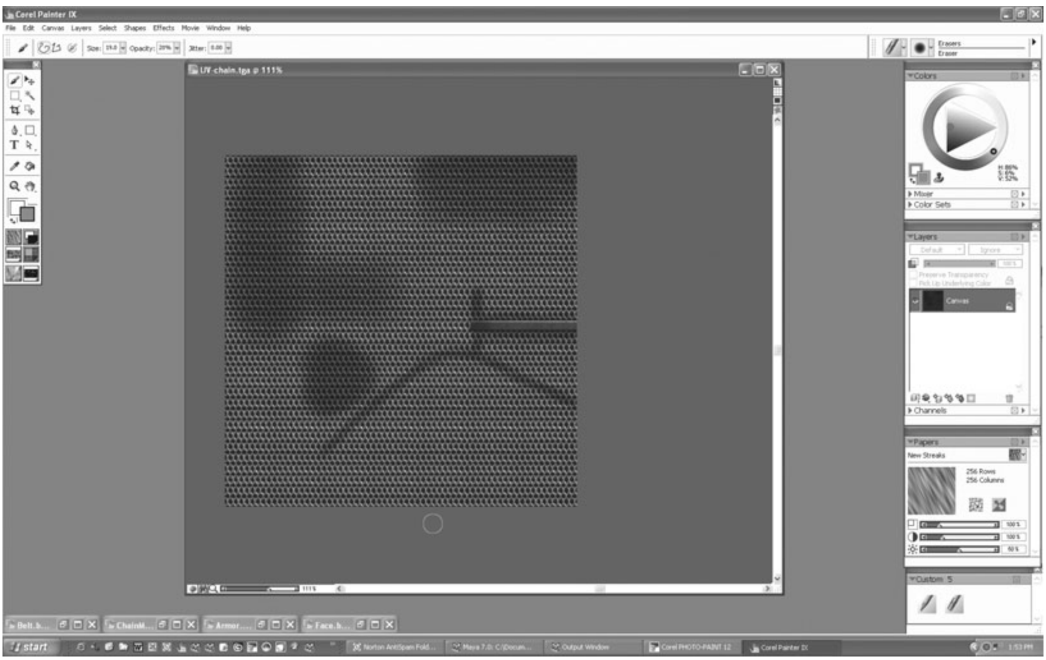


Figure 6.64
Add shadows and a leg strap.

Creating the Belt Texture

The belt texture requires the use a Lambert material because that texture has more of a matte finish.

1. In Painter, open the UV snapshot of the leather and fabric groups.
2. Create a flat leather texture like the one shown in Figure 6.65. Add the leather texture as a layer to the UV snapshot.
3. Work on the straps first, creating a single, long leather strap with a few metal studs, as shown in Figure 6.66.
4. Duplicate the leather strap layer to populate the texture with leather straps, as shown in Figure 6.67.
5. After all the straps have been added, drop all the layers to flatten the image. Make sure you return the Transparency setting of the leather layer to 100 so you don't have the lines of the UV snapshot showing through.
6. Save the image as a BMP file.

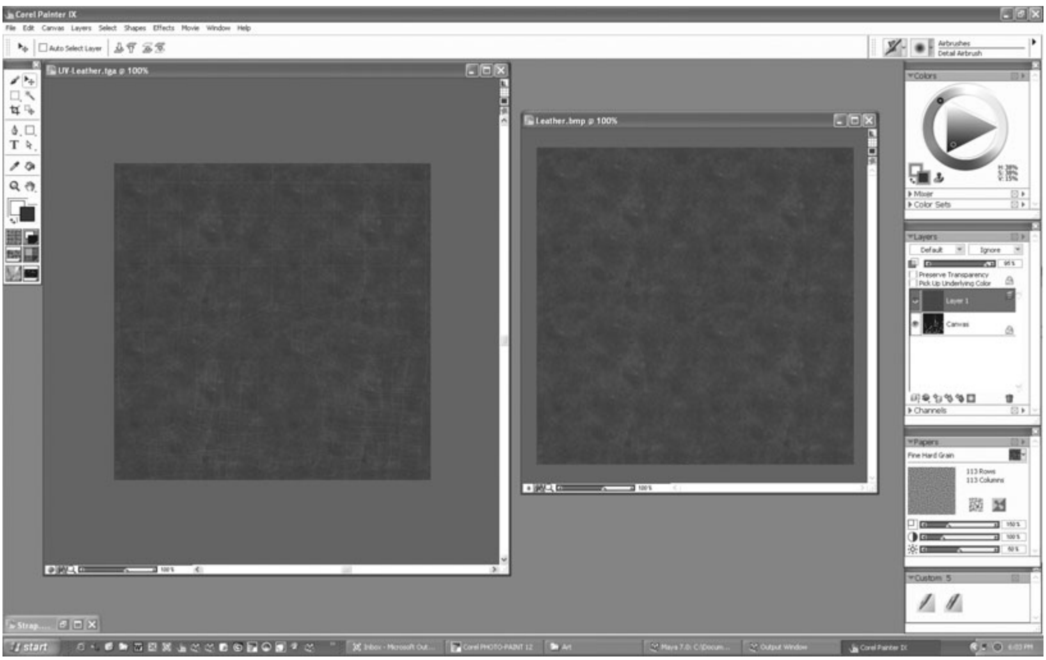


Figure 6.65
Create a leather base texture.

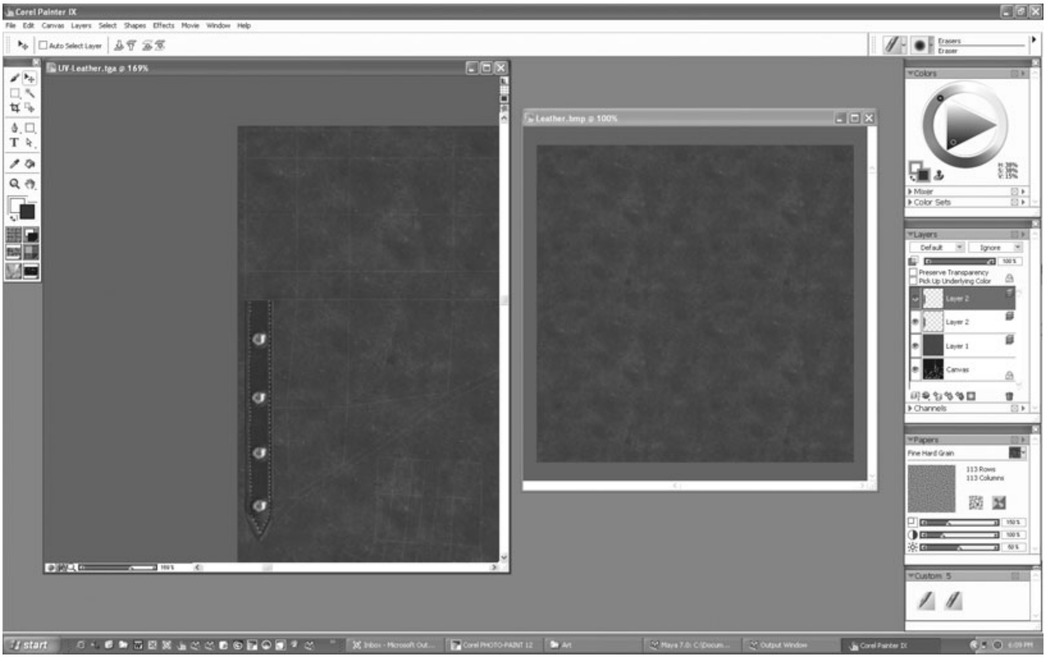


Figure 6.66
Create a single leather strap.

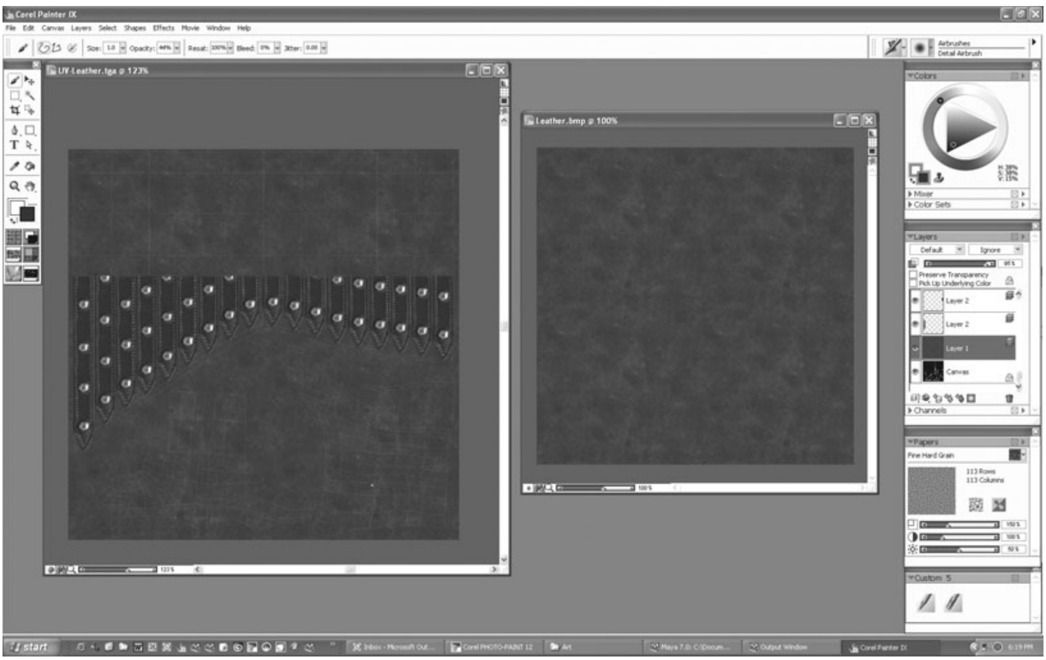


Figure 6.67
Add leather straps going across the texture.

7. Load the UV snapshot into Painter as the bottom layer and the leather layer with the straps as the top layer. Adjust the transparency so you can see the UVs again.
8. Now for the detail on the belt. Open the original leather image and darken it as shown in Figure 6.68.
9. Copy and paste the image and move it down a layer so that it is between the two existing layers and erase the areas that will be in shadow, such as the bottom of the belt and below the straps.
10. When you are finished with the shadow areas, add some highlights to the top of the belt.
11. The scarf and boots are a different color from the leather. Use the Lasso tool to isolate each and change their color using the Adjust Colors setting found in the Tonal Controls submenu of the Effects menu. The adjusted colors should look like what's shown in Figure 6.69.
12. Paint in the details for the scarf and boots as shown in Figure 6.70.

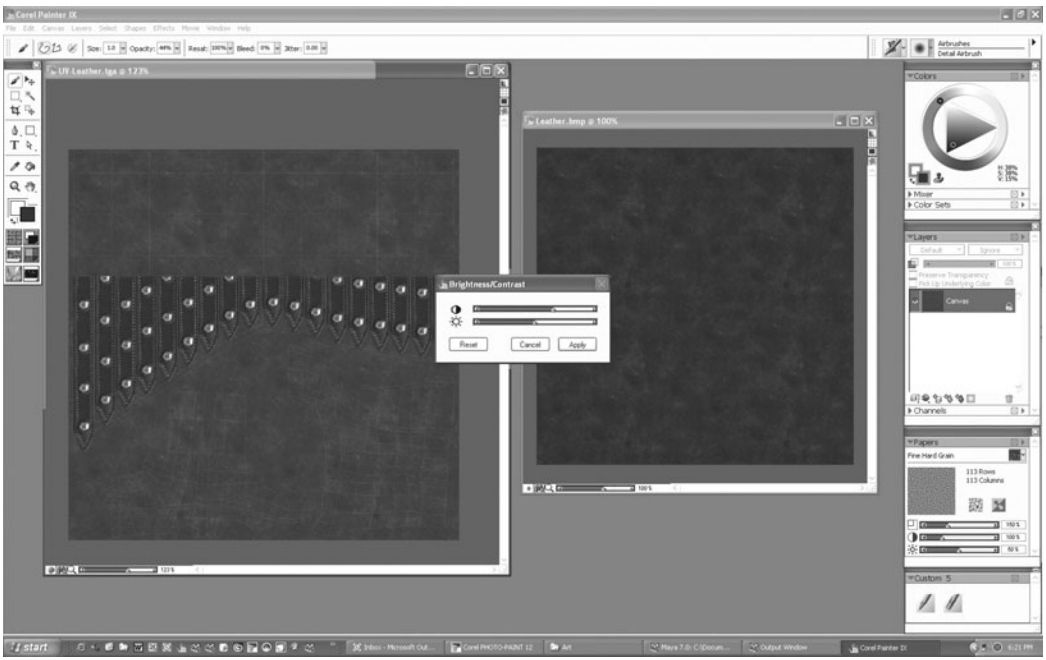


Figure 6.68
Darken the original leather texture.

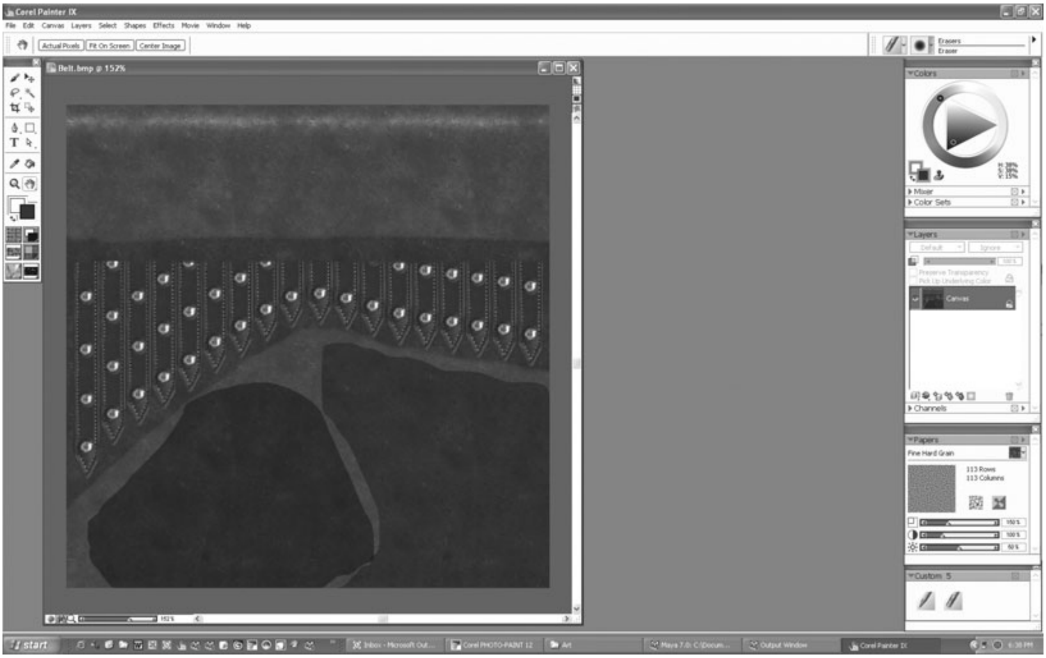


Figure 6.69
Change the colors of the scarf and boots.

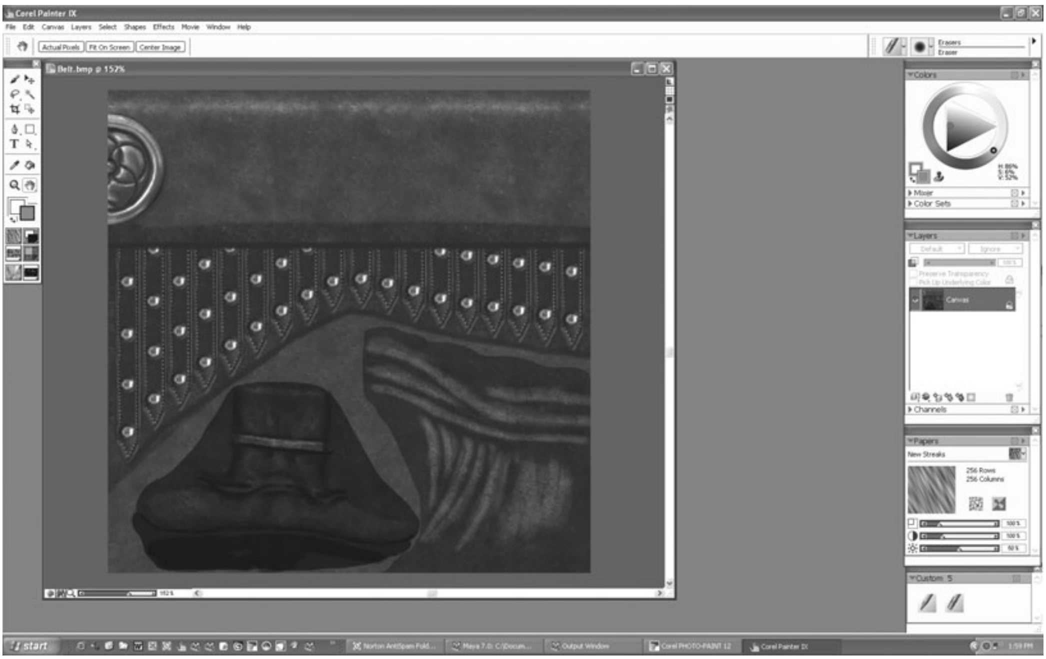


Figure 6.70
Add highlights and shadows to the scarf and boots.

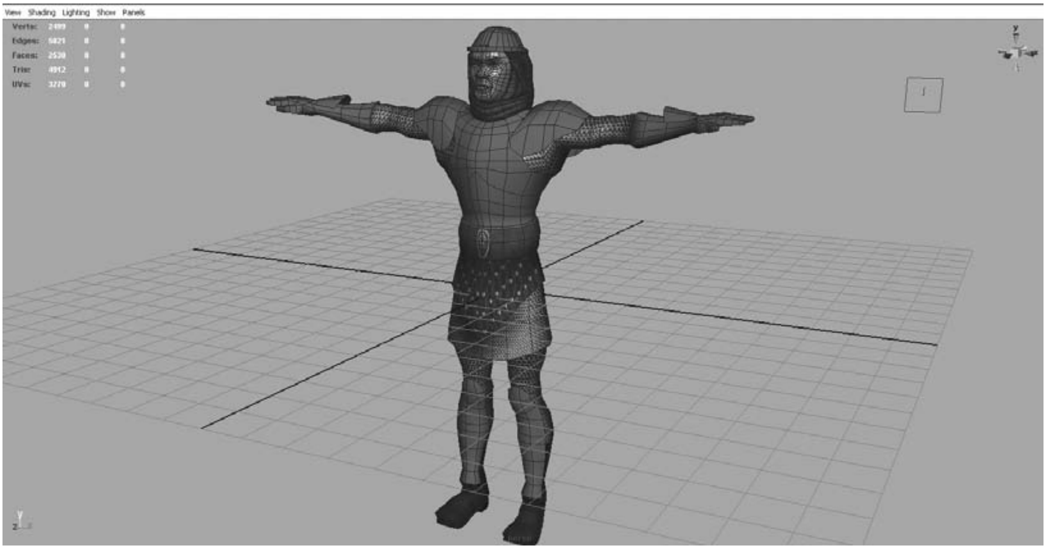


Figure 6.71
Apply the new material to the model to see how it looks.

13. Load the texture into a Lambert material and apply it to the belt UV group as shown in Figure 6.71.

The model is starting to take shape. You just need to do one more texture for the main model: the metal plate armor.

Creating the Plate Armor Texture

The plate armor texture will use a Blinn material to give it the shiny highlights of metal.

1. Start by making a flat color texture that is a light gray-blue (see the black-and-white Figure 6.72 shown here, or the full-color version of the image found on the CD).
2. Lay the gray-blue texture over the UV snapshot of the plate armor UVs.
3. Create a new image that has etchings and boundary lines for decorations on the armor as shown in Figure 6.73.
4. Copy and paste the decorations as a new layer on top of the layer stack and make it transparent so you can see through to the lower layers while still retaining the detail (see Figure 6.74).

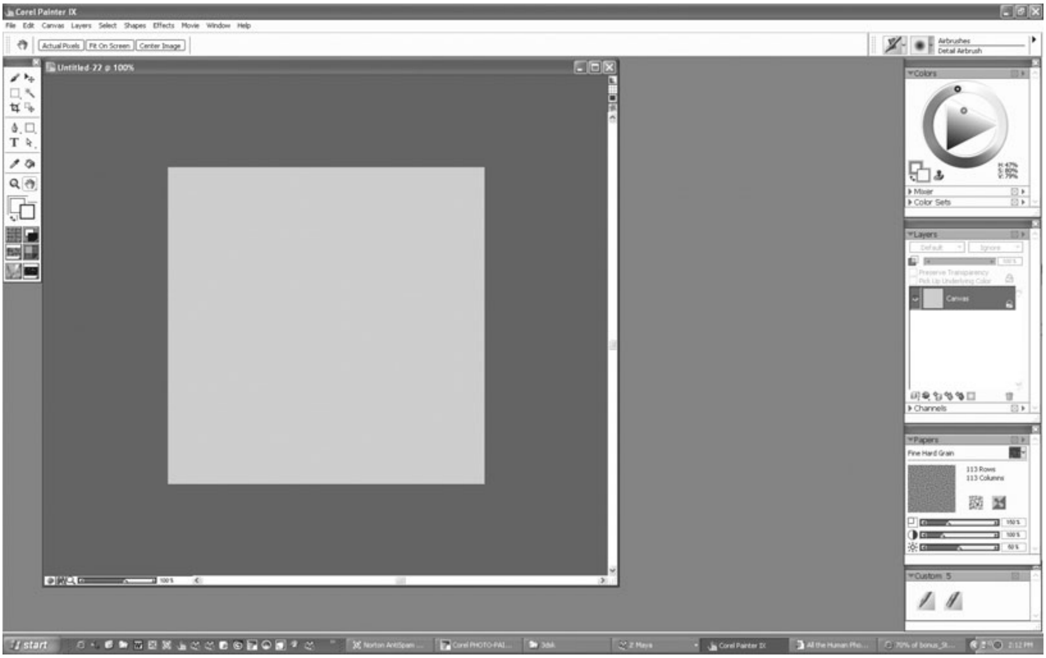


Figure 6.72
Create a light gray-blue color texture.

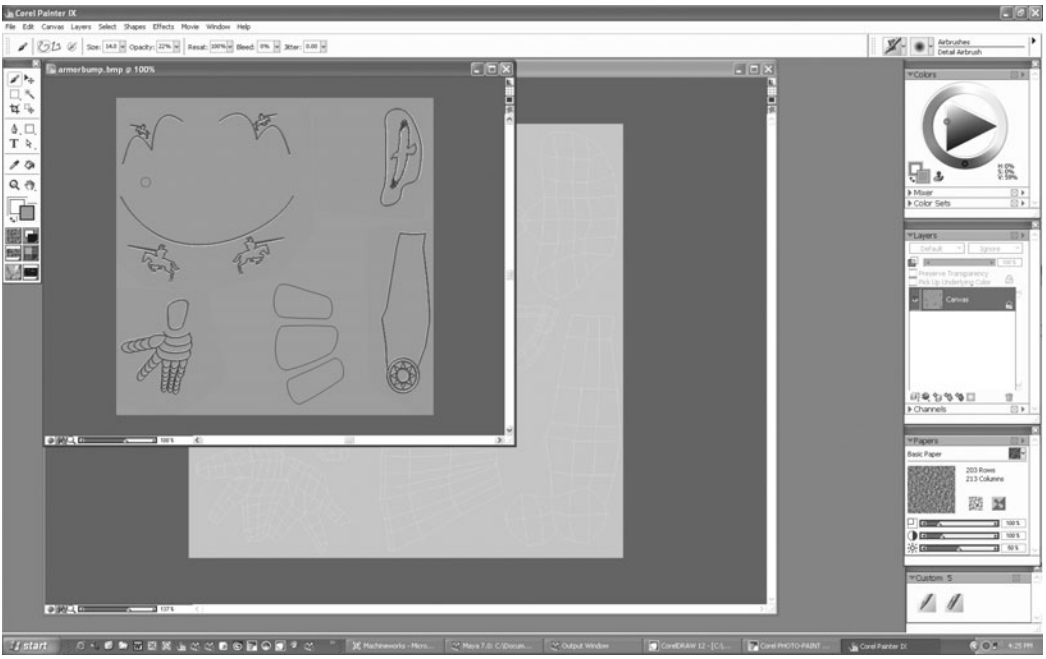


Figure 6.73
Draw the decorations that go on the armor.

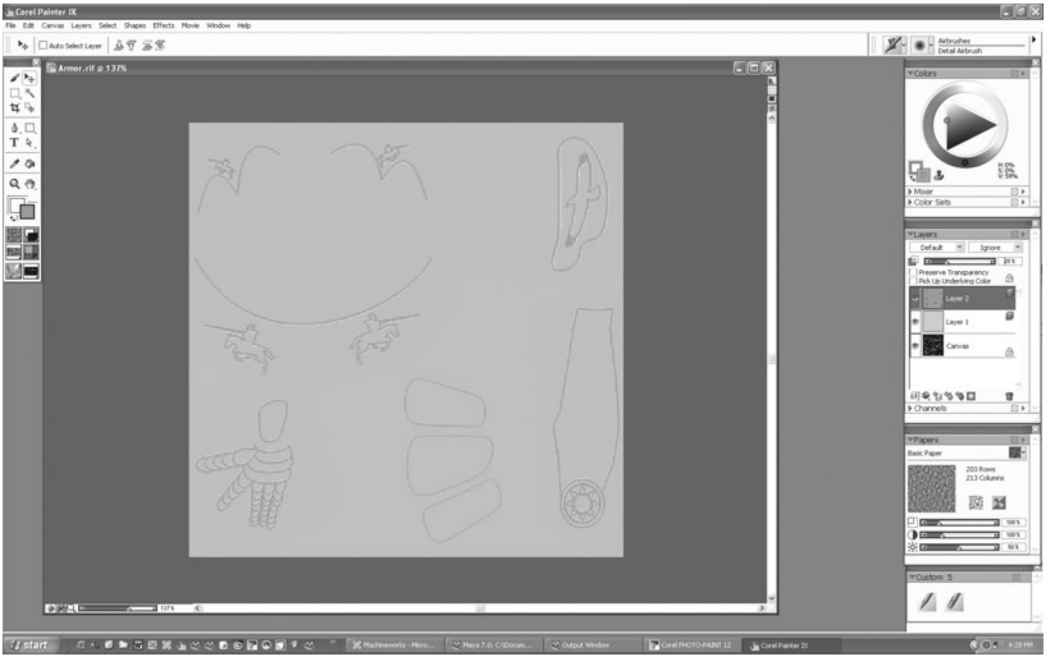


Figure 6.74
Add a decoration layer.

5. Add a yellow-orange layer for the brass portions of the armor. Place it between the gray-blue and decoration layers.
6. Erase brass areas of the texture as shown in Figure 6.75.
7. Use the Magic Wand selector on the top layer to select the gray areas around the decoration lines. Make sure you have the Contiguous checkbox unchecked so the Magic Wand selector selects all of the gray areas.
8. Press Ctrl + X to remove the gray; this will brighten the underlying colors.
9. Remove all the transparence and drop all the layers so you can work on a single layer.
10. Add a little shading around the plates on the gantlet and a dark color for the underside.
11. Because you started off with a simple color for the texture, it is a little flat. Add a surface texture to give it a very slight variation as shown in Figure 6.76. Use the New Streaks paper and increase the softness to spread the texture out.

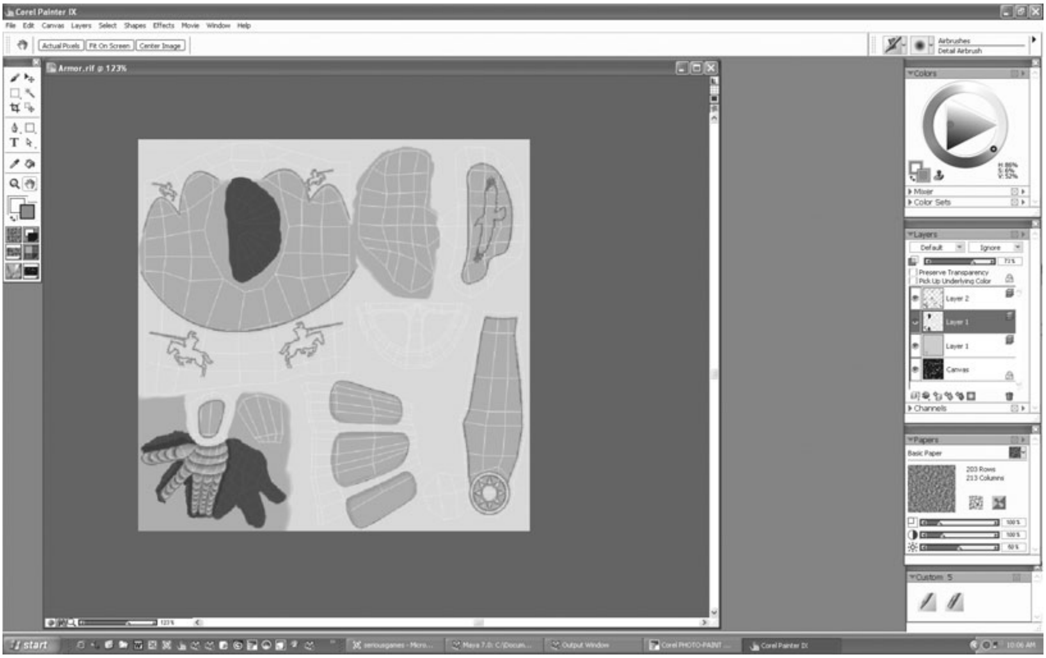


Figure 6.75
Erase the brass areas according to the decoration boarders.

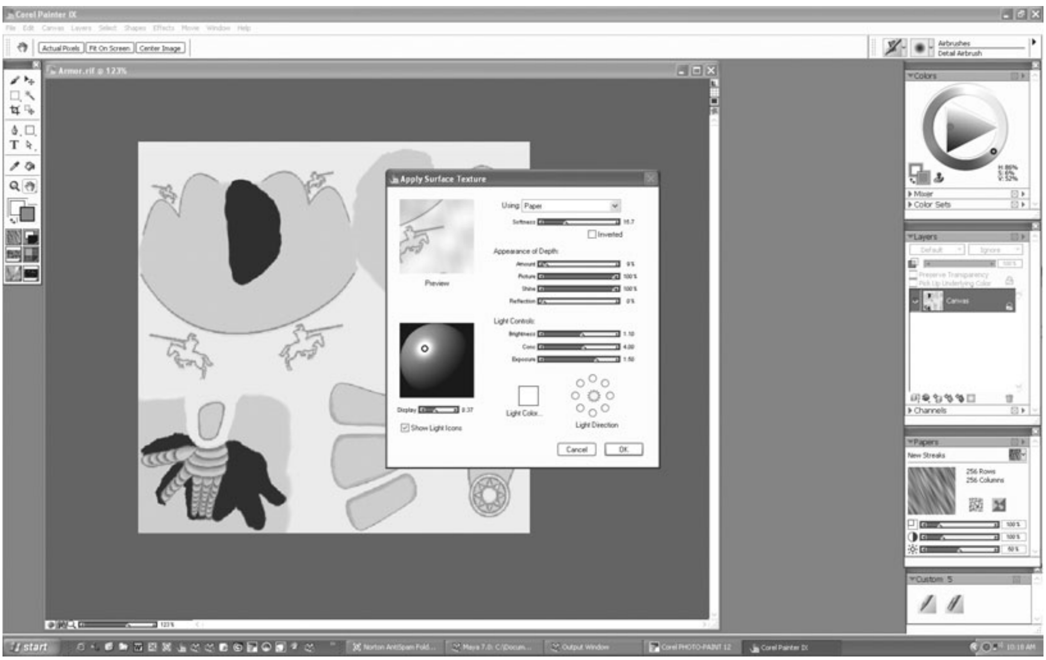


Figure 6.76
Add a surface texture to the underlying texture.

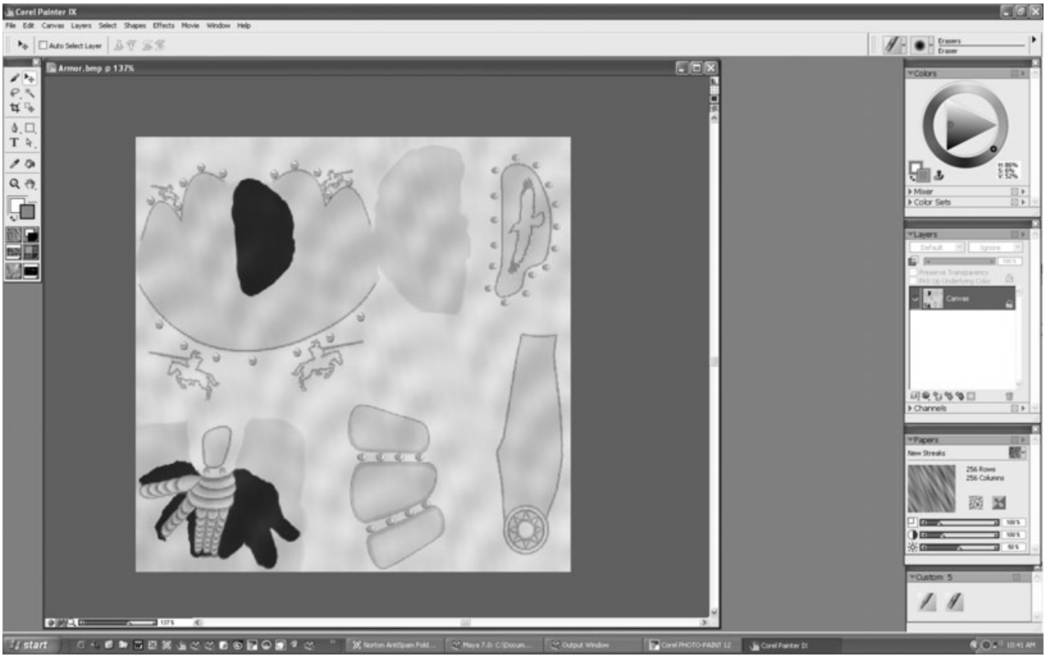


Figure 6.77
Create rivets around the borders of the armor.

12. Add rivets around the breast plate, shoulder plate, and wrist guard, as shown in Figure 6.77.
13. You also want to add some rivets to the helmet, but you may need the UV snapshot for a guide. The rivets should look like those in Figure 6.78.
14. Load and apply the new texture to the model.

The main model is now completely textured. In Figure 6.79, I did a quick test render to see how he looks. If you your game engine available, you might want to export the model and take a look at him in the engine. There is no substitute for seeing you work in action in a game.

Creating the Eye Texture

The eye is a shiny object and as such should be constructed from a shiny material. You can have the eye texture share a texture with one of your shiny materials. Your model has two shiny materials—the chainmail and the plate armor—but finding a good texture for the eye will be easier on the plate armor, so use that material. You could create a separate texture for the eye, but doing so for such a

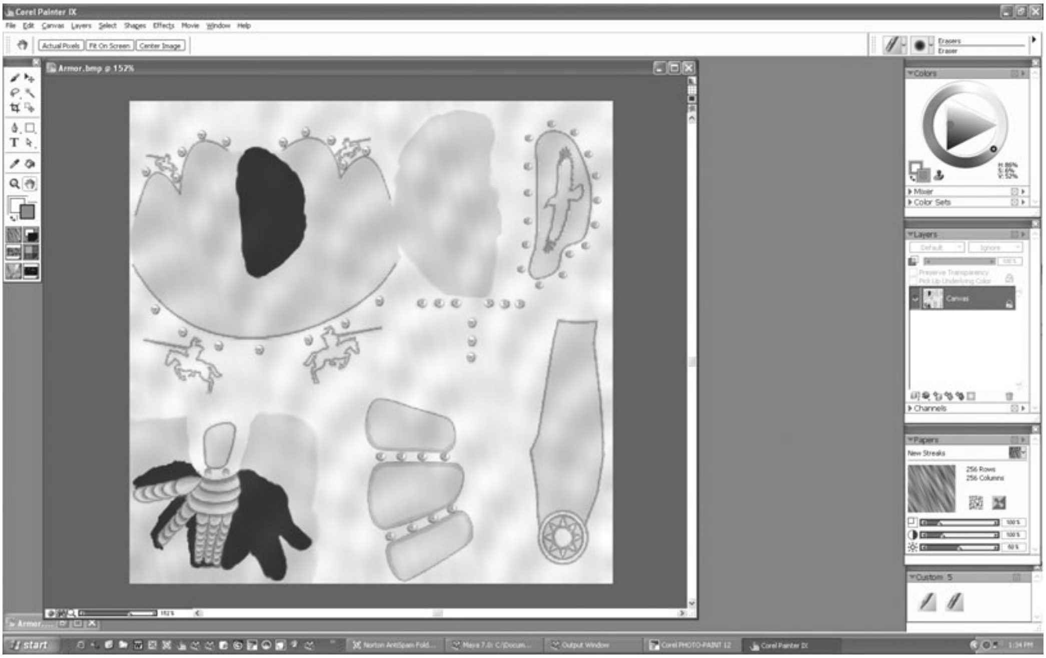


Figure 6.78
Add some rivets to the helmet.

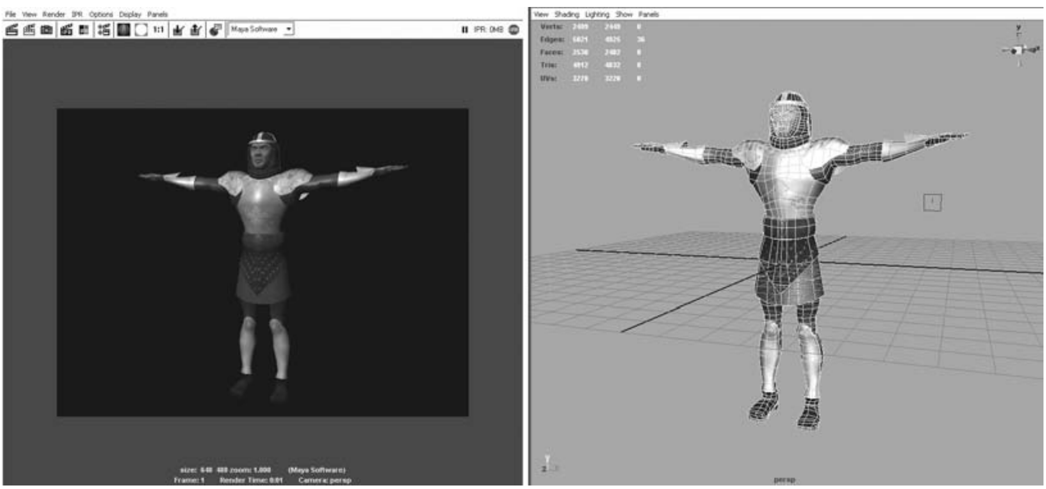


Figure 6.79
Check your model out in a test render.

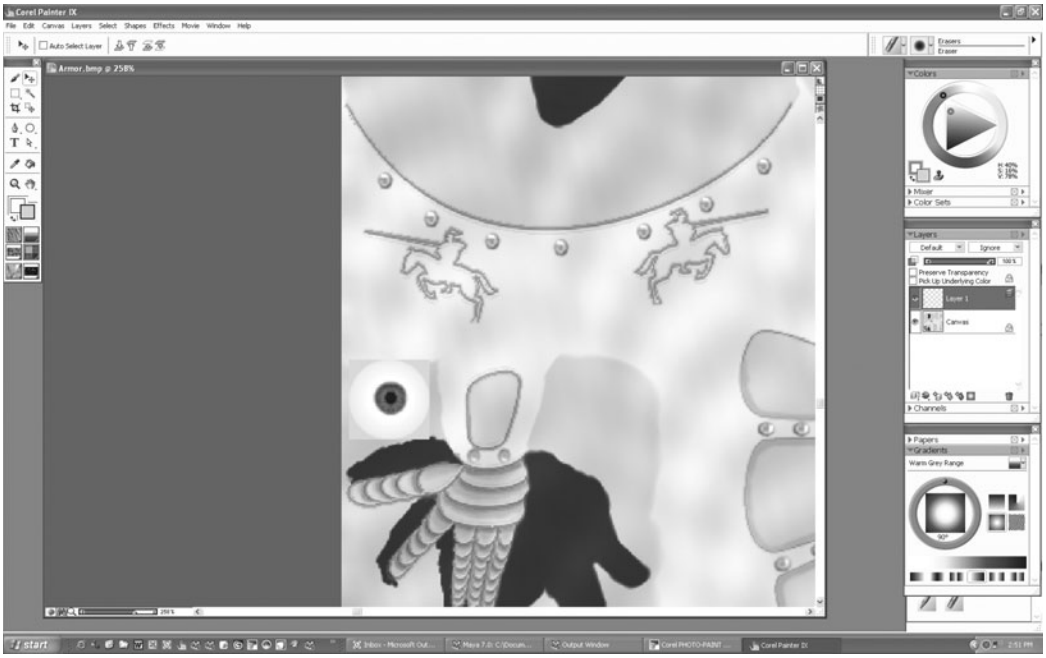


Figure 6.80
Place the eyeball in an unused area of the palate armor texture.

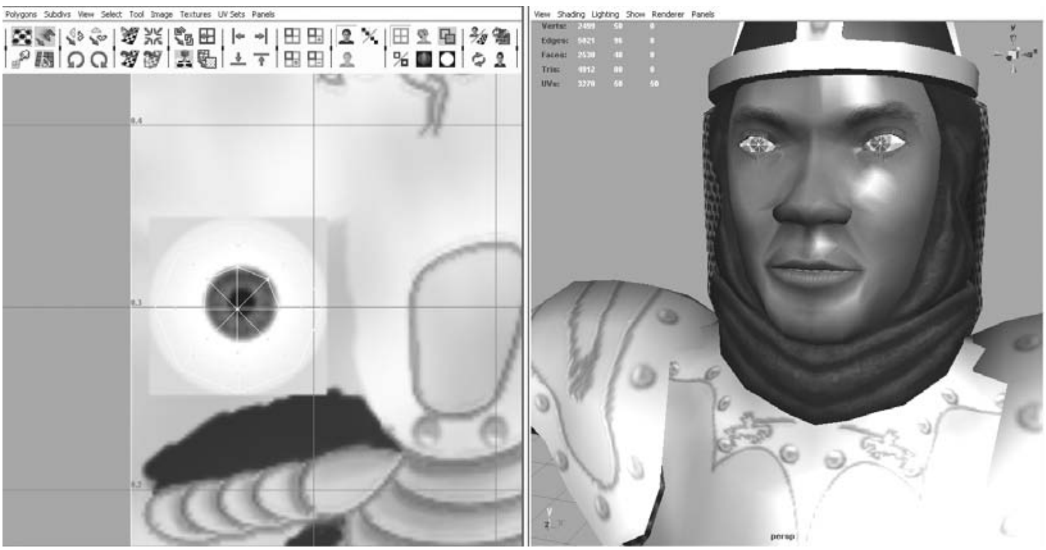


Figure 6.81
Scale and move the UVs to fit over the eye.

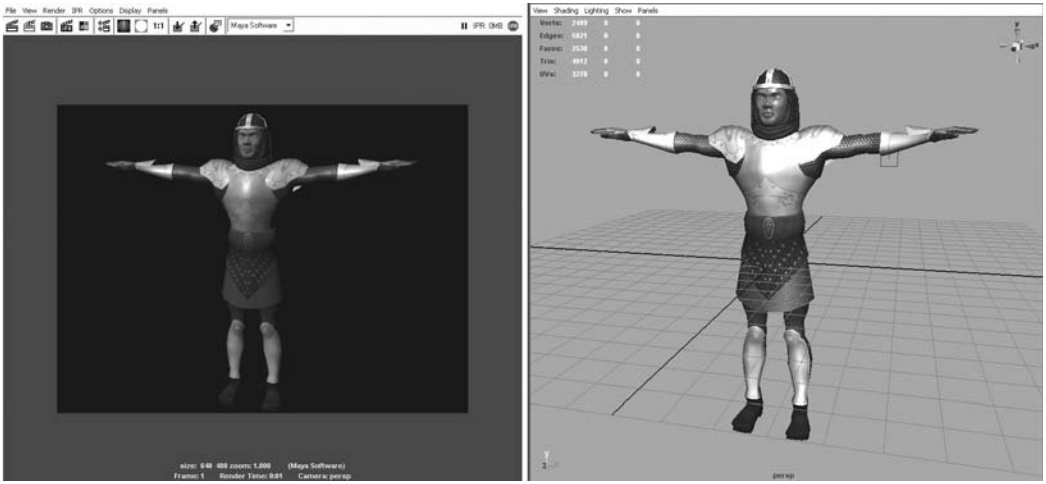


Figure 6.82
The model is now ready for rigging.

small aspect of the model seems like a waist of resources. I placed the eyeball just above the gantlets, as shown in Figure 6.80.

Now all you need to do is apply the plate armor texture to the eyeball models and then scale the UVs and move them to the eyeball on the texture, as shown in Figure 6.81.

The model is now textured and ready to be animated. Figure 6.82 shows my textured, finished model. Compare it to yours to see how you did. If your game engine requires all textures to be in the 0 to 1 texture area, you may need to move the UVs. It will make the UVs harder to select but it might make the rendering go faster.

Summary

As you learned in this chapter, getting the right textures in the right places on a model is a complex operation! From a technical standpoint, the most demanding task is organizing the UVs. Artistically, the creation of the texture maps is probably the most demanding. Following are the topics covered in this chapter:

- Using Automatic Mapping
- Common tools in the UV editor
- Finding and combining UV pieces

- Creating UV groups
- Creating texture groups
- Creating a face texture
- Creating a pattern texture
- Using layers in creating textures
- Creating materials
- Applying textures to UV groups
- Testing the model using test renders

Chapter 7

Character Rigging

Before a character can be animated, it must to have a way to move its various body parts.

Right now, your character is just a polygon mesh with textures. Sure, it looks nice, but all it can do is stand there with its arms out. Technically speaking, you could probably animate your character by moving vertices with the modeling tools and keying the frames, but that would be tedious and time consuming to say the least.

In order to expedite the animation process and simplify the programming of characters in games, a character is attached to a skeletal frame of bones and joints called a *skeletal system*. Using a skeletal system, the artist can rotate joints that influence the vertices of the model, causing them to move in much the same way that our bodies move with our own skeletons. The process of creating a skeleton and attaching the vertices to the skeleton is referred to as *rigging the model*. In this chapter, you will learn how to build a skeleton for you character model and then attach the model to the skeleton.

External Movement Versus Internal Movement

Before I get started on the specifics of building a skeleton, I want to talk about the difference between external movement and internal movement. When characters animate in a video game, they move in two ways: internally and externally. *External movement* is when the location in a scene changes. For example, if a character moves from the front of a castle to the edge of a clearing, external movement has occurred. *Internal movement* is when any of the character's body parts or any of the objects with which the character is interacting change positions. For example, if the character raises his arm to shield his eyes from the glaring sun, the movement of the arm is internal movement. It is also internal movement if an object has moving parts, like the wheels on a car or the handlebars on a bicycle.

External movement is often controlled by the game code and, with help from the game artist to ensure the character's external movement matches his internal movement, is set up by the game programmer. Internal movement, one the other hand, is primarily the realm of the game artist and is often referred to as *character animation*. Note that internal movement and external movement often happen at the same time, such as when a character is walking.

Skeletal Systems

As I mentioned, a *skeletal system* is a system of joints and bones used for animating characters or other objects in games that have internal moving parts. The system is hierarchal, meaning that higher joints in the hierarchy, often called *parents*, influence joints lower in the hierarchy, often called *children*. If, for example, a shoulder joint rotates, the joints of the arm move based on the shoulder rotation; put another way, the shoulder joint is the parent of the elbow joint and the elbow joint is the child of the shoulder joint. The parent of all the joints in a skeletal hierarchy is called the *root joint* and is usually placed in the model's pelvis. The root joint is the base joint for animating all other joints in the skeletal system. Some game engines use an extra joint, placed at the origin, or 0 in the x, y, and z coordinates, as the root joint.

Building the Skeleton

When building the skeleton for your model, it is very important to place the joints in the right areas of the model to get the animation to look right. If a joint is placed incorrectly, it will influence the model incorrectly, causing the model's movement to look odd or awkward. In most cases, if you are familiar with human anatomy, you will be able to build a good skeleton just by following the joints in your own body. You can place a knee joint in the knee area of your model in the same location that the knee joint in your own body exists. In fact, most of the names of the joints you will place in your model will have the same names as the joints in your own body.

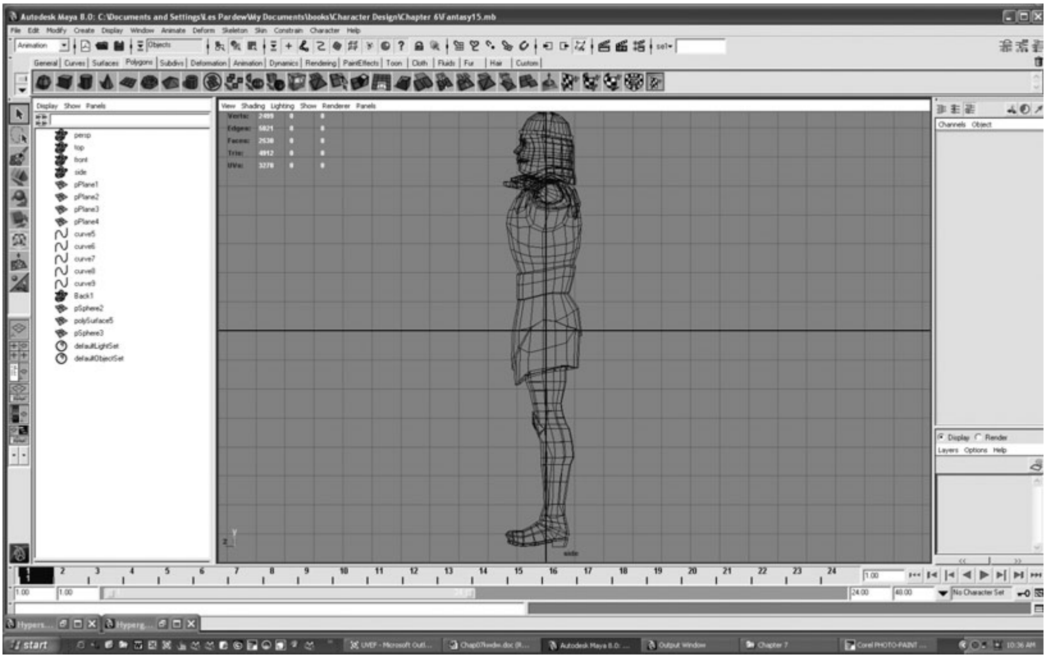


Figure 7.1
Start in side view.

Bring up the character model you created in the last chapter to get started creating a skeleton.

1. Choose Animation from the drop-down list in the Status line of the main menu to switch to the Animation main menu.
2. Change your view to the Two Panes Side by Side layout, with the Outliner on the left side of the screen. (The Outliner, accessed via the third Quick Layout button from the top, is handy for seeing the construction of your character's skeleton.)
3. Switch to the orthographic side view, as shown in Figure 7.1.

Note

By default, Maya places the joint along the origin line; for this reason, it will place a joint on 0 in the x and y axis when you are working in the side view. When you're working in the front view, the joints will be place on 0 in the z and y axis. When in the top view, joints will be placed on 0 in the x and z axis. If you try to place a joint in Perspective view rather than Orthographic view, there is no telling where the joint might end up! For this reason, it is a good practice to place joints only in one of the three orthographic views so you know where your joints will end up.

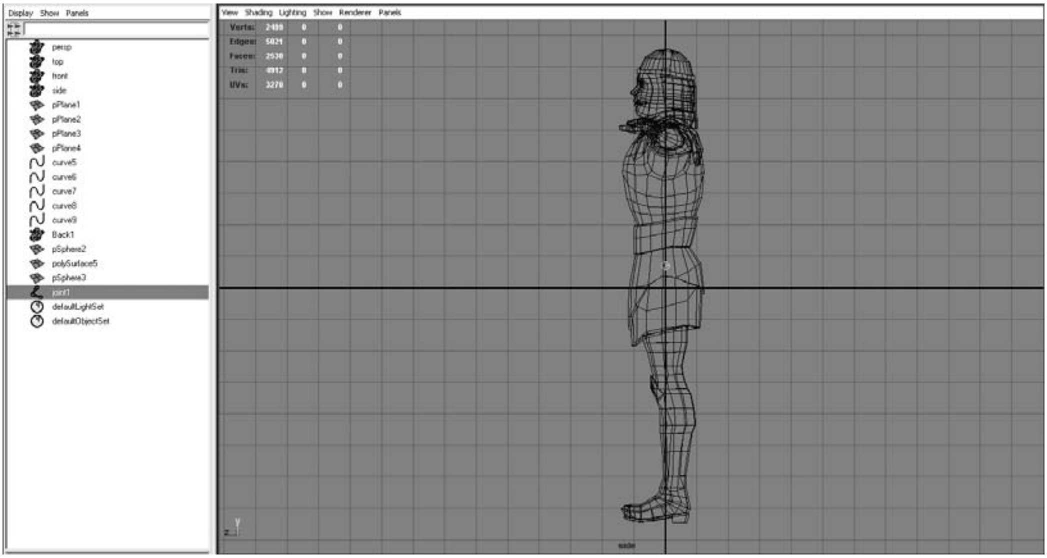


Figure 7.2
Place the first joint in the character's pelvis.

4. Open the Skeleton menu on the Animation main menu and select Joint Tool.
5. Click in the model roughly where the character's pelvis is. As shown in Figure 7.2, a green circle representing the joint will appear.
6. Place three more joints going up the character's spine to the base of the neck, as shown in Figure 7.3. This is called a *joint chain*. Notice that as you place each joint, a bone will connect it to the joint before it. (Bones are represented by diamond-shaped objects, with the larger end closer to the parent joint and the point next to the child joint.)
7. You need to name each joint either at the end of the building process or as you go. Because naming as you go helps to eliminate confusion later, I strongly recommend it. To name a joint click on it and, in the Channel box, type a name for the joint (I chose Pelvis.).

Note

Names can also be changed in the Outliner or in the Hypergraph.

8. Name the three joints you added in step 6 Spine01, Spine02, and Spine03 respectively. (Notice that the name of each joint also changes in the Outliner; if you click the + symbol next to the joint in the Outliner, you can see the children of that joint.) Figure 7.4 shows the Pelvis joint and its three children.

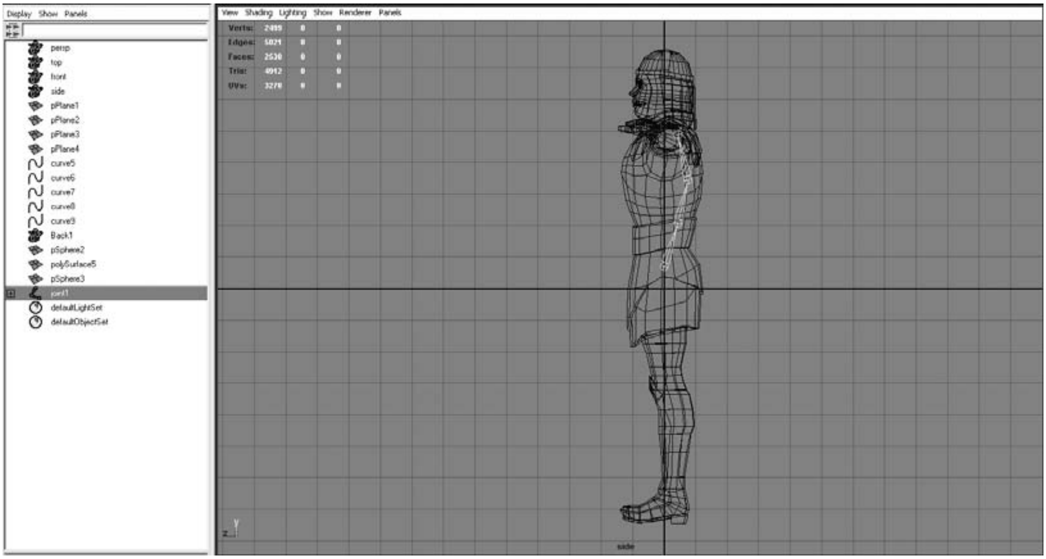


Figure 7.3
Place the joints for the spine.

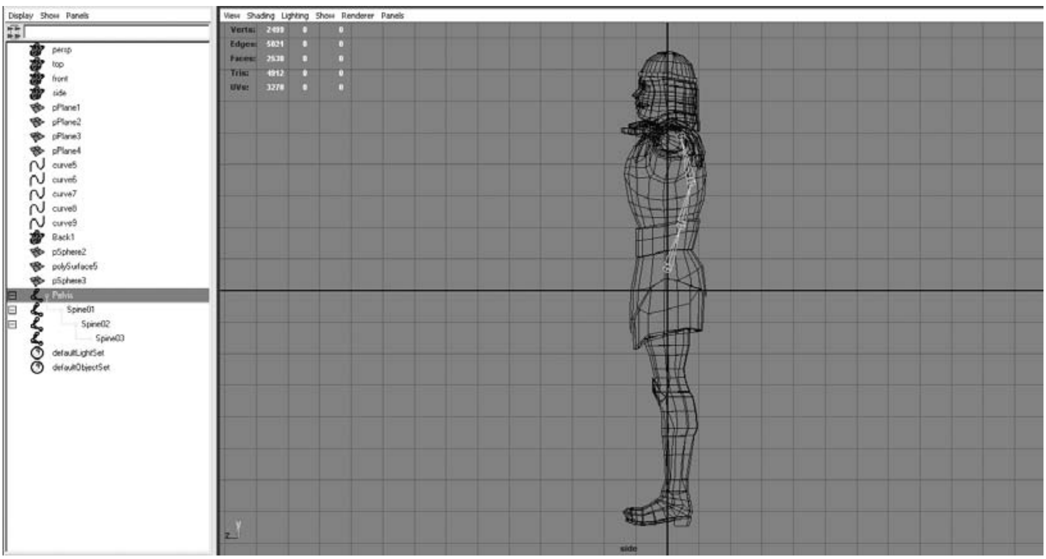


Figure 7.4
The Outliner shows the children of the Pelvis joint.

9. Next, create the joints of the leg. Place the first joint in the hip and subsequent joints in the knee, ankle, and the ball of the foot, as shown in Figure 7.5. (You only need to build the joints for one side of the skeleton because you'll mirror the joints later.)

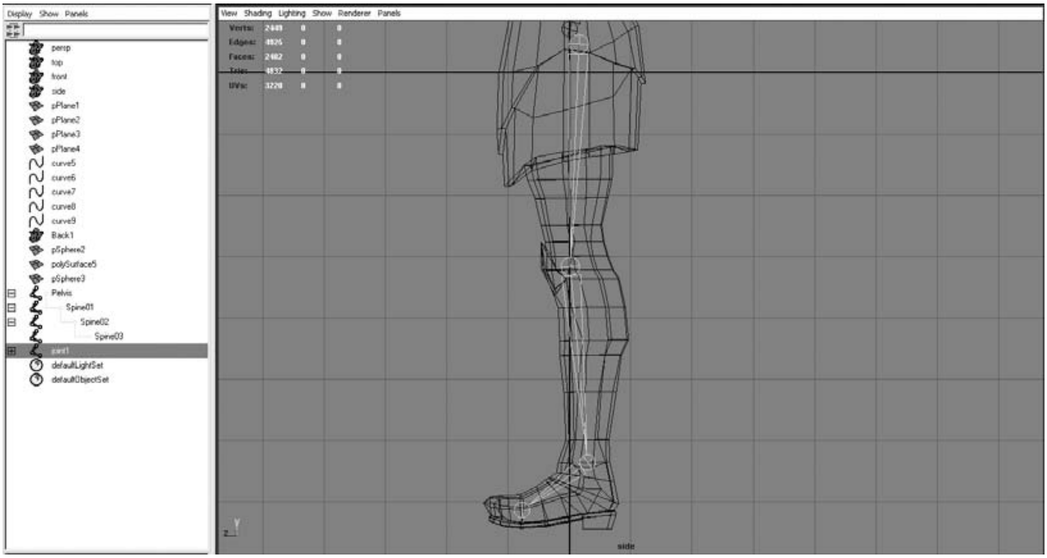


Figure 7.5
Place the joints of the leg.

10. Name the joints Hip_L, Knee_L, Ankle_L, and Foot_L.

Note

When naming joints, distinguish the joints of the left side of the character's body from those of the right side. In this example, I indicate that a joint is on the left or right side by appending _L or _R to the end of the joint's name, respectively.

11. Switch to the front orthographic view and, starting at the hip and working downward, move and rotate the joints to fit in the character's left leg as shown in Figure 7.6. (Using the Move and Rotate tools in this scenario is similar to using them to move and rotate geometry.) Notice how when you move the Hip_L joint, all the other joints follow.

12. Connect the Hip_L joint to the Pelvis joint, with the Pelvis joint acting as the parent. To do so, select the Hip_L joint; then, while pressing the Shift key down, select the Pelvis joint. When both joints are selected, press the P key on your keyboard. A bone will appear between the two joints, as shown in Figure 7.7.

Note

When "parenting" a joint—that is, making one joint the parent of another joint—always select the joint that will act as the child first and joint that will be the parent second.

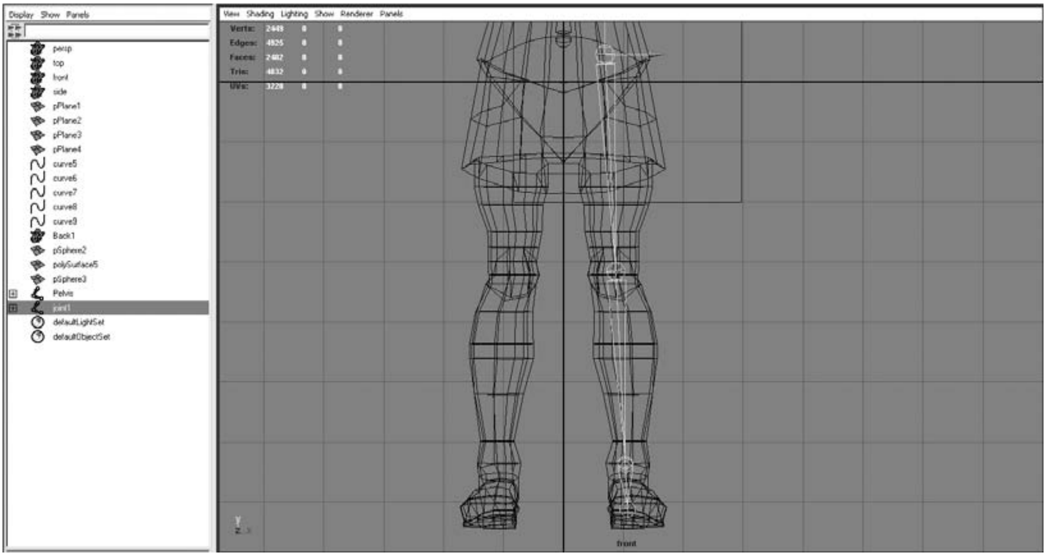


Figure 7.6
Move the leg joints into place in the front view.

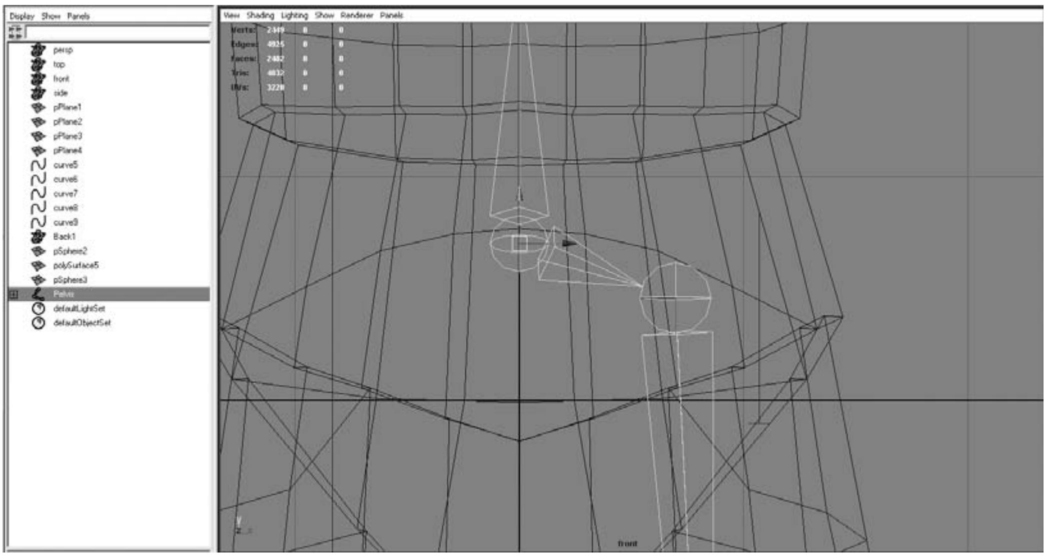


Figure 7.7
Connect the Hip joint to the Pelvis joint.

13. Next, create the joints of the arm and the hand. In front view, place a joint in the shoulder, the elbow, the wrist, and the three knuckles of the fingers as shown in Figure 7.8. Name these joints Shoulder_L, Elbow_L, and Wrist_L, I_Finger01_L, I_Finger02_L, and I_Finger03_L, respectively.

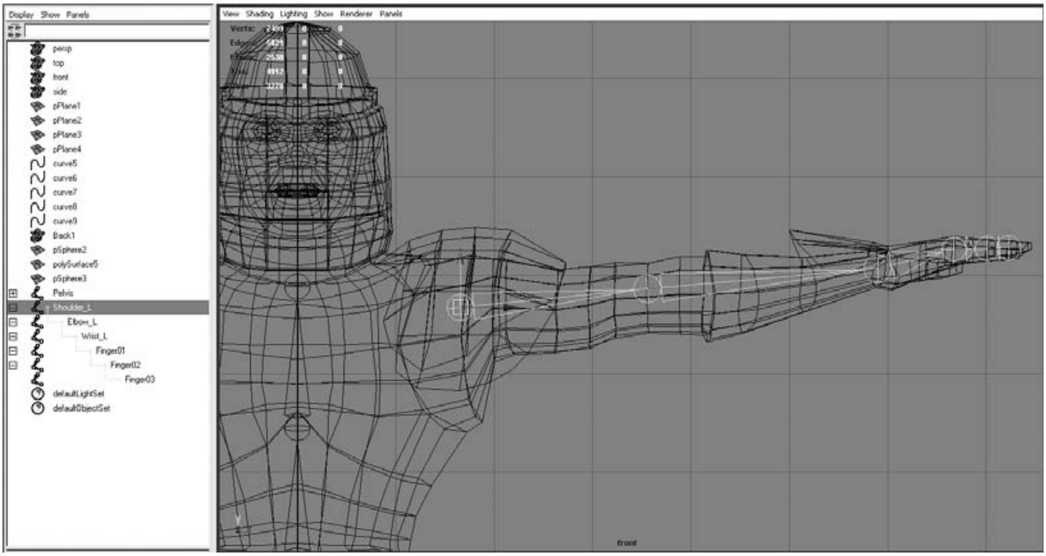


Figure 7.8
Place the joints for the arm and hand.

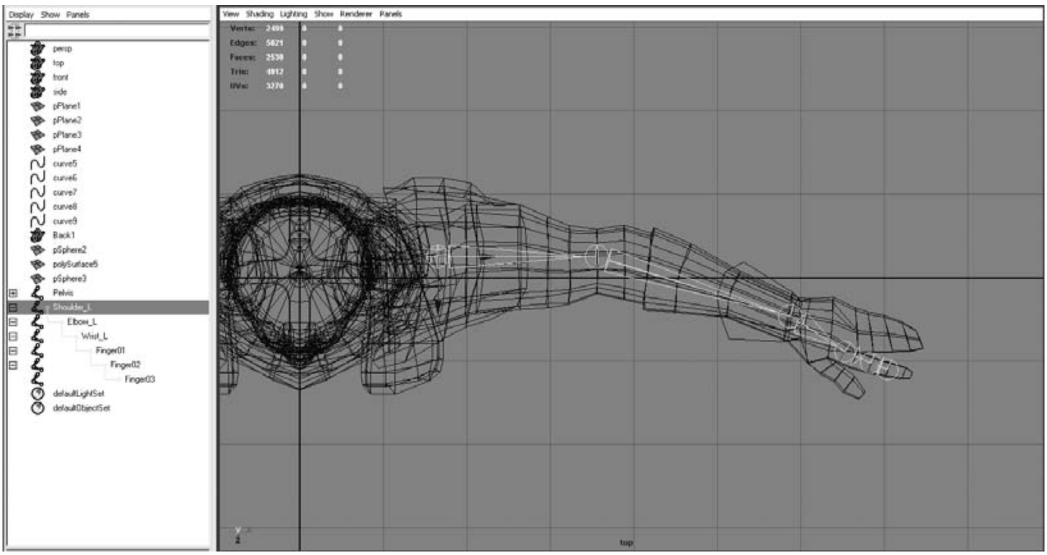


Figure 7.9
Position each joint correctly in top view.

14. Switch to top view and move and rotate the joints you just created to fit the arm, as shown in Figure 7.9.

15. Rather than creating new joints for the other fingers and thumb, you can duplicate the existing finger joints and move them into place as shown in

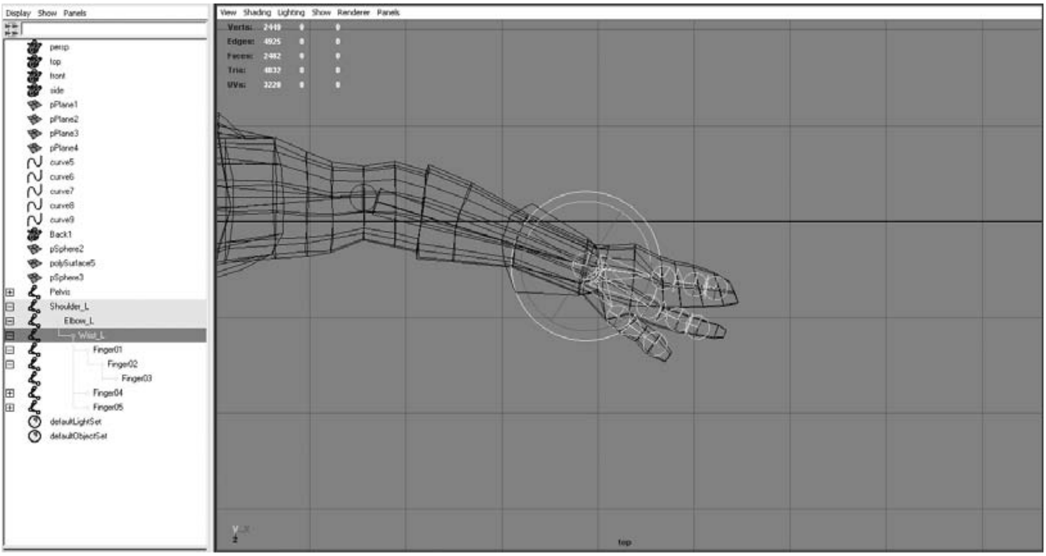


Figure 7.10
Duplicate the finger joints and move them into position in the hand.

Figure 7.10. To do so, simply duplicate the first knuckle in the finger chain (I_Finger01_L) by pressing Ctrl+D on the keyboard. This automatically duplicates all its children and connects it with the Wrist_L joint.

16. Connect the Shoulder_L joint to the top joint of the Spine03 joint, as shown in Figure 7.11.
17. Switch back to side view and track in on the head. Place a joint for the neck, the head, and the chainmail toward the back of the head, as shown in Figure 7.12. Name the joints Neck, Head, and Chainmail, respectively.
18. Next place one joint in the jaw and one in the eyeball as shown in Figure 7.13. Name the joints Jaw and Eyeball_L, respectively.

Note

Because this character will have only simple facial expressions such as opening and closing his mouth and blinking his eyes, the number of joints you need to place is minimal. For complete facial animation, you would need to place additional joints for the eyebrows and cheeks.

19. Connect the Eyeball_L joint to the Head joint, with the Head joint acting as the parent. Repeat with the Jaw joint.

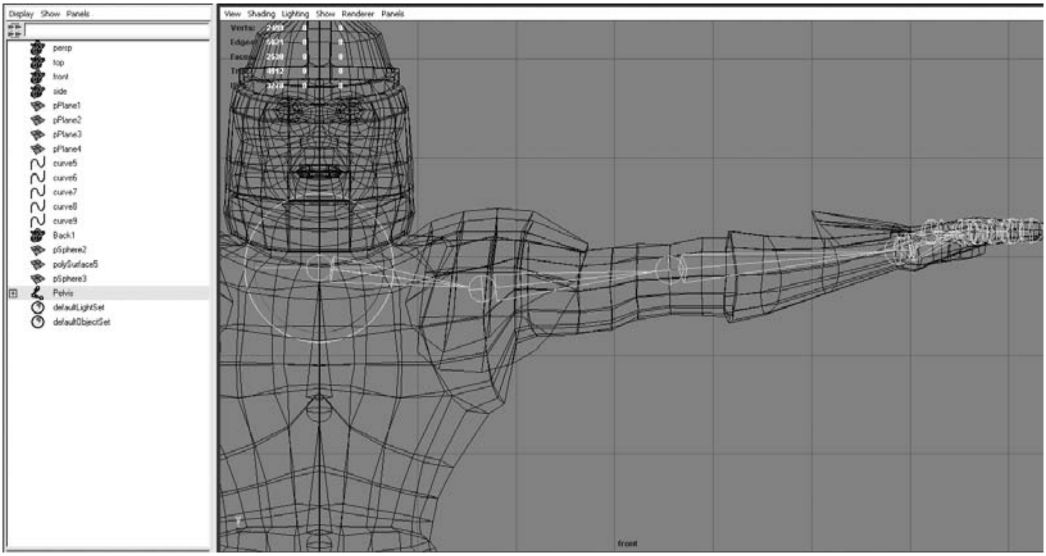


Figure 7.11
Connect the shoulder and arm joints to the spine.

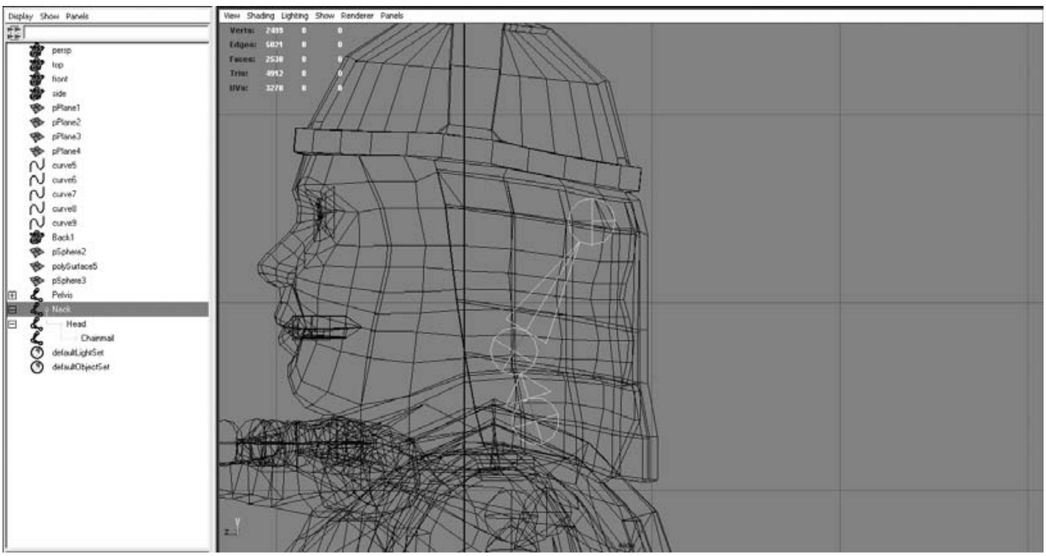


Figure 7.12
Place the joints of the head.

20. Switch to front view and move the Eyeball_L joint into position over the eye as shown in Figure 7.14.

21. Use the Mirror Joint function to mirror the Eyeball_L joint in order to create the character's other eye. (Not only will the Mirror Joint function

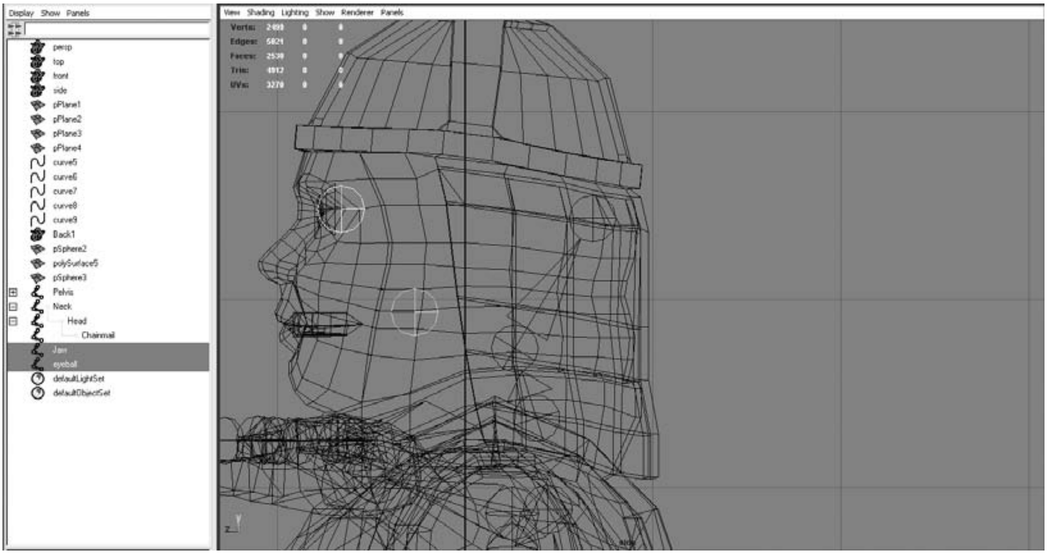


Figure 7.13
Place joints for the eyeballs and the jaw.

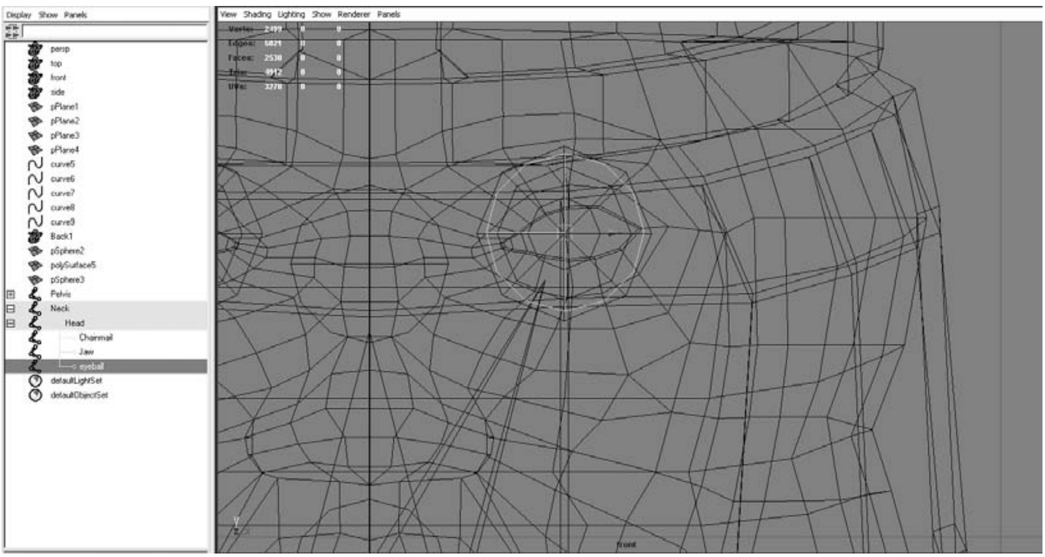


Figure 7.14
Move the eyeball joint to the eye.

create a mirror of the existing eyeball joint in the other eye, it will also copy that joint's name.) To begin, open the Skeleton menu in the Animation main menu and select Mirror Joint to open the Mirror Joint Options dialog box.

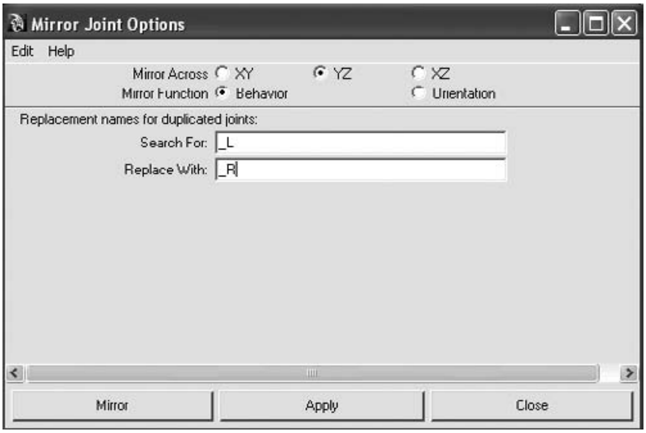


Figure 7.15
Mirror the Eyeball_L joint to the other eye.

22. Select the YZ option button in the Mirror Across section and the Behavior option button in the plane. Type _L in the Search For field and _R in the Replace With field. Select the Eyeball_L joint and click the Apply button in the dialog box to mirror the joint (see Figure 7.15).

23. The eyelids need to move independently from the eyeballs. Otherwise, the character will not be able to blink. For this reason, you need to have duplicate joints in exactly the same place as the two eyeball joints. Duplicate each eyeball joint and rename the joint Eyelid_L and Eyelid_R respectively.

Inverse Kinematics

As you learned, the hierarchical nature of the skeleton system enables you to position joint chains by rotating the parent joints of the chain. This type of animation is called *forward kinematics*. Sometimes, however, you may want to simply grab the child joint—say, a hand—and place it where it should go. The problem is, with forward kinematics, if you move a character's hand to a new location, the elbow joint doesn't follow—resulting in a distorted forearm.

To solve this problem Maya uses *inverse kinematics*, which allows the animator to move a child joint and have the parent joints follow. When applied to a joint chain, Maya creates a handle called an *IK handle* on the child joint and a structure connecting it to its parent joints.

In this section, I show you how to attach an IK handle to the arms and legs of your character.

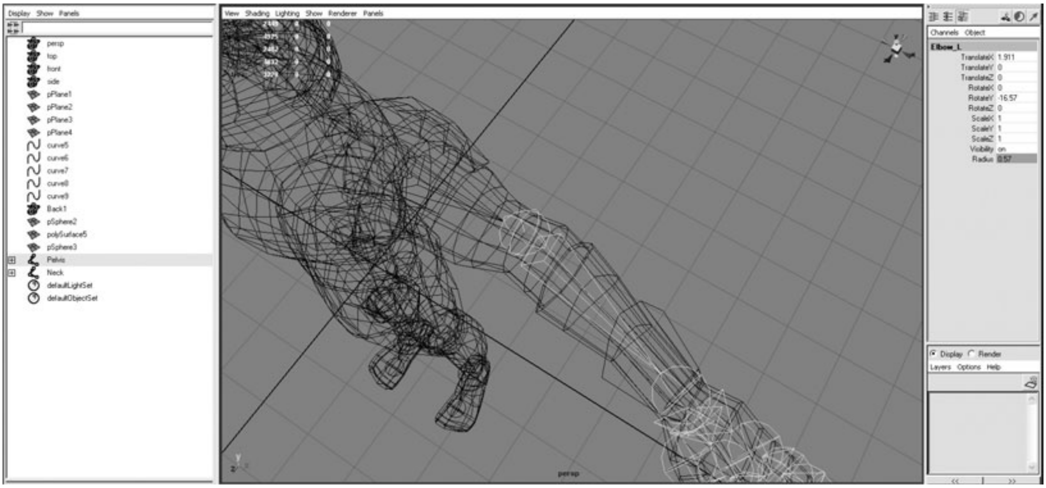


Figure 7.16
The elbow needs to be rotated in the y axis.

1. Before you can attach an IK handle to a joint chain, you must set a preferred angle, which tells Maya which direction you want the joint to rotate. In this example, you need to set a preferred angle for the left elbow joint. To begin, click on the Elbow_L joint (see in Figure 7.16). Then, in the Channel box, check the Rotate Y setting to determine whether the elbow is already rotated. If it isn't, rotate it at least −10 degrees.

Note

A preferred angle tells Maya what axis you want the joint to rotate in. It is an easy way to keep the arms and legs rotating correctly while using inverse kinematics.

2. Select the Shoulder_L joint. Then, open the Skeleton menu in the Animation main menu and select Set Preferred Angle, as shown in Figure 7.17.
3. The arm is ready for you to apply inverse kinematics. To begin, open the Skeleton menu in the main Animation menu and select IK Handle Next, click the Shoulder_L joint and then the Wrist_L joint. The IK handle is applied to the joint chain, as shown in Figure 7.18. To use the IK handle, click the L-shaped lines extending out from the Wrist_L joint.
4. The Knee_L joint is not rotated, but it needs to be to set a preferred angle. To rotate it 10 degrees in the z axis, type `10` in the Rotate Z setting in the Channel box, as shown in Figure 7.19.

Figure 7.17
Set the preferred angle of the elbow joint.

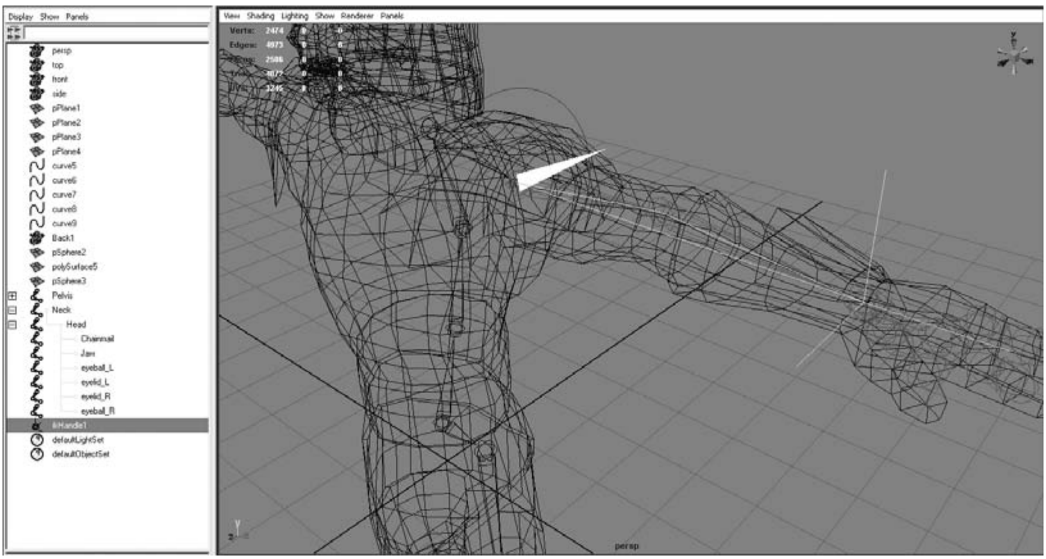


Figure 7.18
Apply an IK handle to the arm.

5. Open the Skeleton menu in the Animation main menu and choose Set the Preferred Angle
6. Rotate the Knee_L joint back to 0 in the z axis.
7. Apply an IK handle to the leg as shown in Figure 7.20.

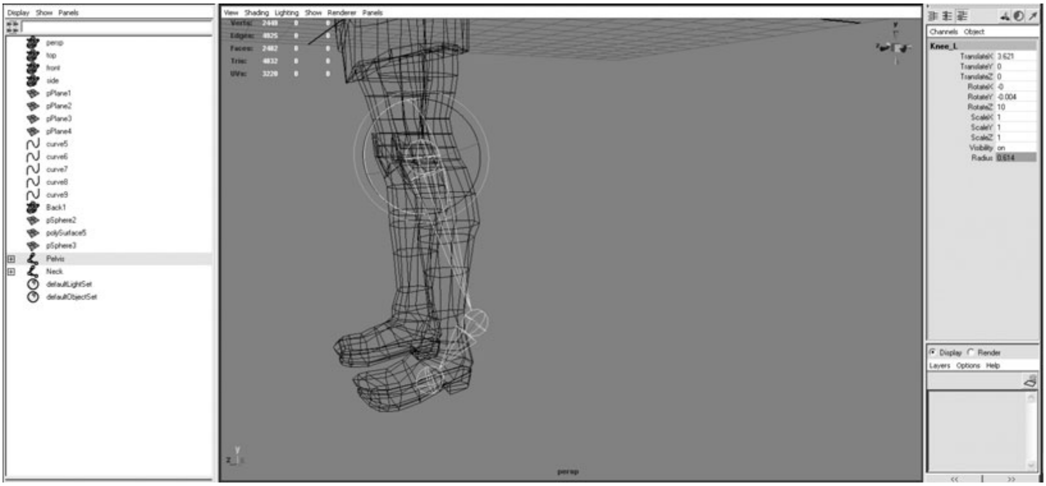


Figure 7.19
Rotate the Knee_L joint 10 degrees in the z axis.

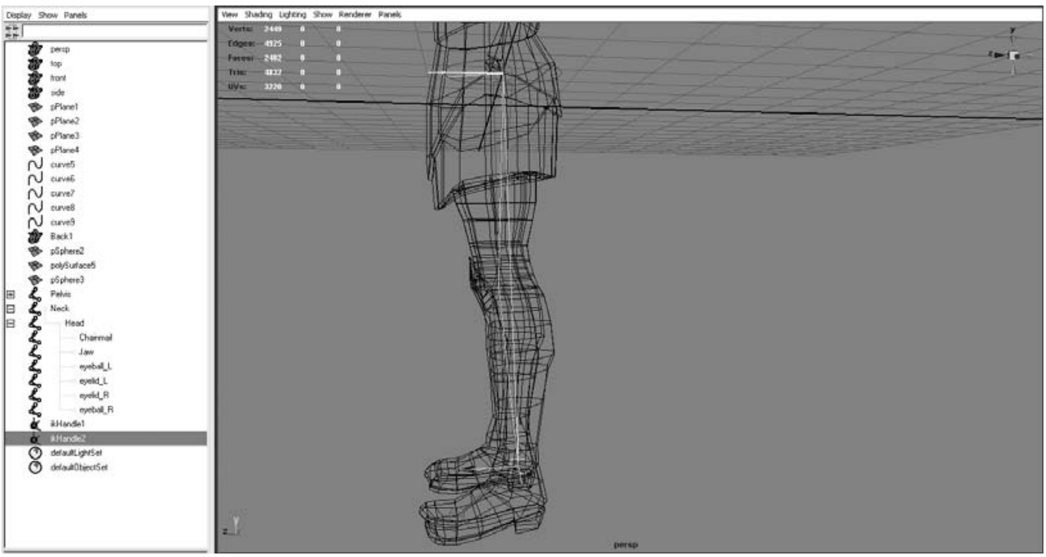


Figure 7.20
Apply an IK handle to the leg.

Finishing the Skeleton

Now that you have the hang of adding inverse kinematics, you're ready to finish building your skeleton. This involves mirroring the side you've already done.

1. Following steps 21 and 22 in the section "Building the Skeleton, mirror the joints in the legs as shown in Figure 7.21.

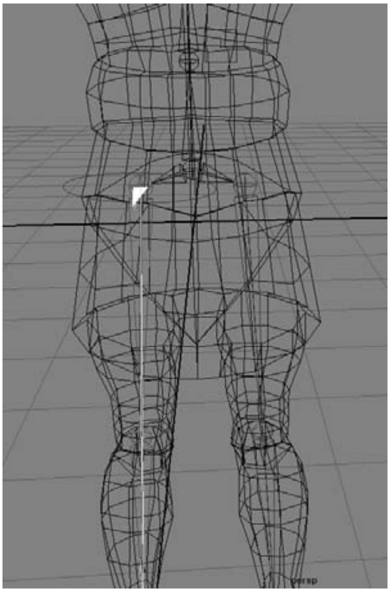


Figure 7.21
Mirror the leg joints.

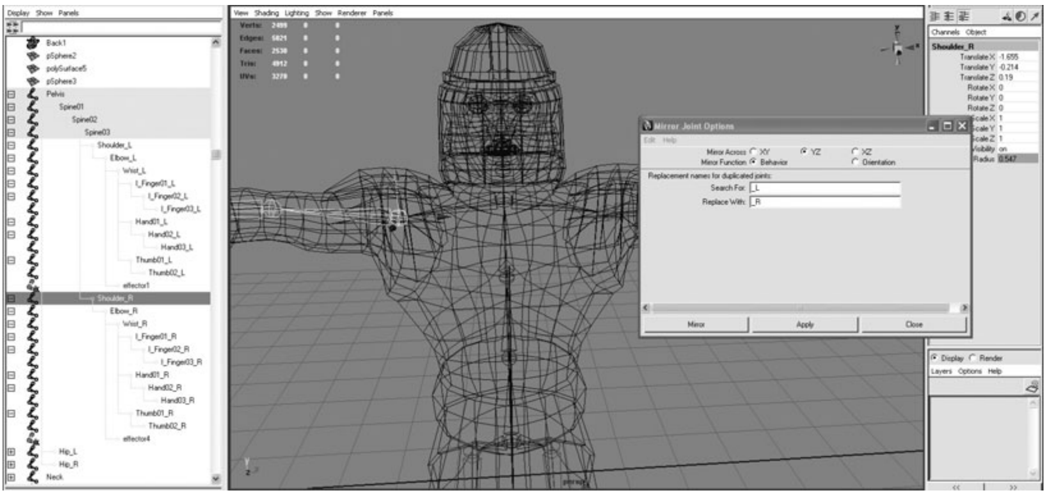


Figure 7.22
Mirror the arm joints.

2. Mirror the joints in the arms as shown in Figure 7.22.
3. Connect the Neck joint to the Spine03 joint, as shown in Figure 7.23.
4. Just a few more joints and the skeleton will be complete. Create a joint for the shoulder armor as shown in Figure 7.24, naming it Armor_L.

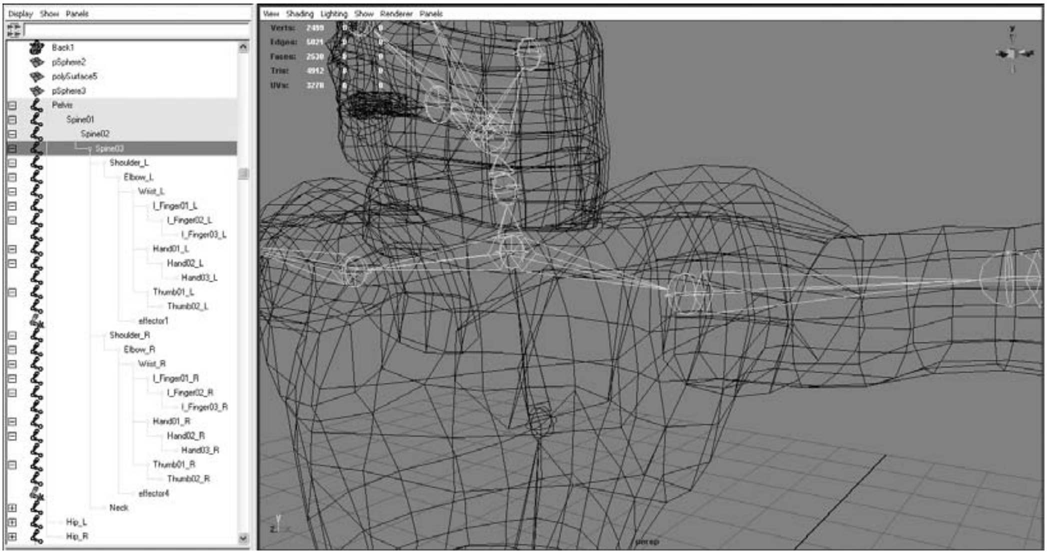


Figure 7.23
Connect the neck to the spine.

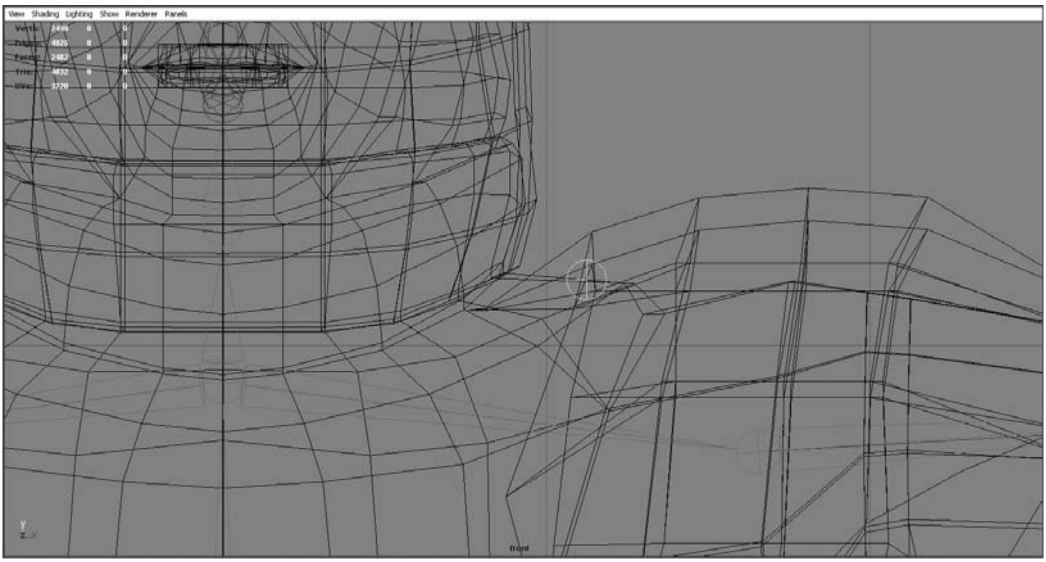


Figure 7.24
Place a joint for the shoulder armor.

5. Mirror the Armor_L joint.

6. Now you need to create two new joints for animating the chainmail at the back of the character's head. To begin, place a joint inside the head about the level of his mouth on the left side and name the joint Chainmail_L.

7. Connect the Chainmail_L joint to the Chainmail joint at the back of the character's head.

8. Mirror the Chainmail_L joint as shown in Figure 7.25.

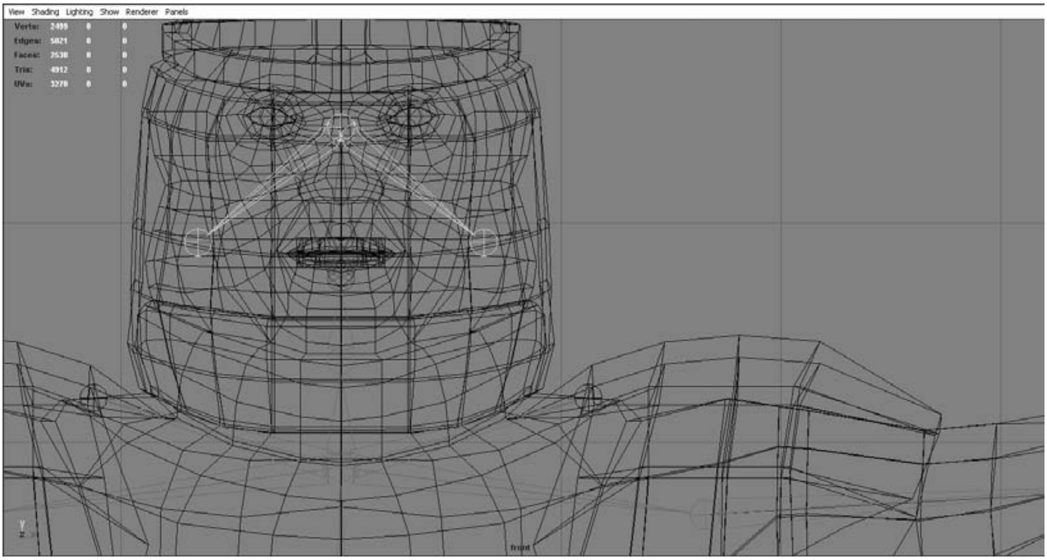


Figure 7.25
Create joints for the animation of the chainmail.

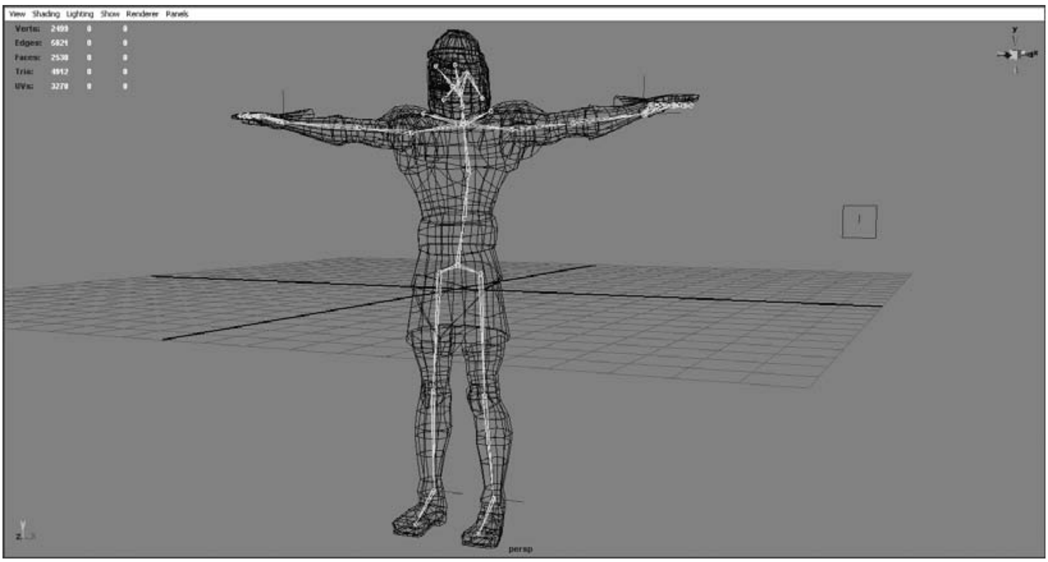


Figure 7.26
The figure should now have a complete skeleton.

9. Rotate the Chainmail joint so the two children joints (that is, Chainmail_L and Chainmail_R) are positioned around its edge in the side view. The skeletal system now has all the joints it needs for animation; it should look like the one in Figure 7.26.

The skeletal system is just about ready to be attached to the model, but one more thing needs to be done: The IK handles need to placed in the overall skeletal hierarchy. As shown in the hypergraph in Figure 7.27, they are currently sitting outside of the hierarchy. (To access the Hypergraph, open the Windows menu and choose Hypergraph.)

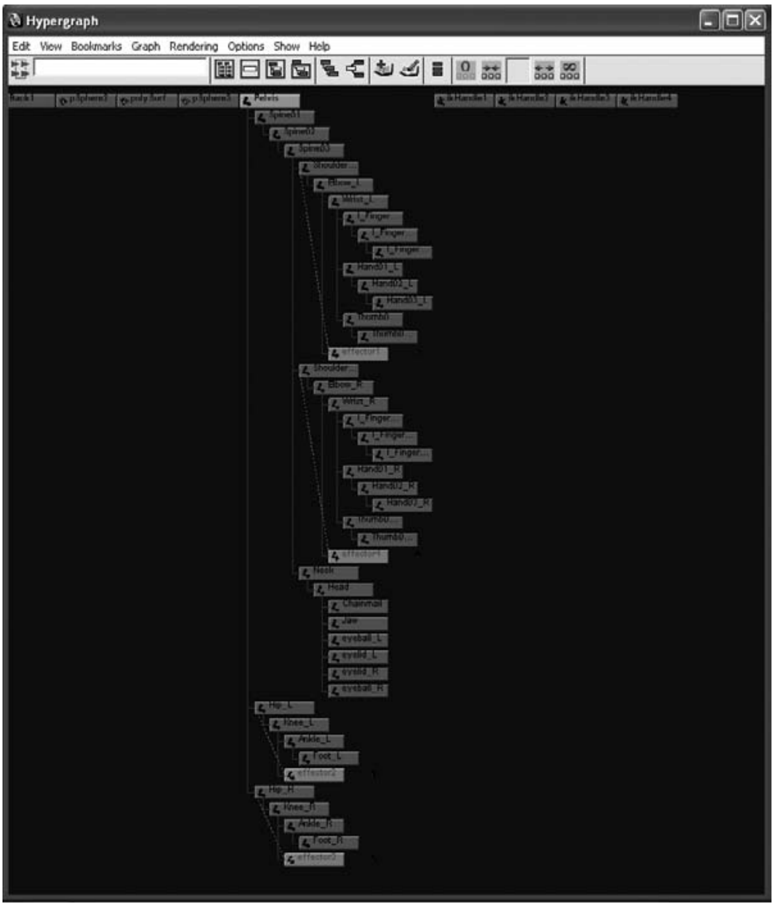


Figure 7.27
The IK handles are sitting outside of the skeletal hierarchy.

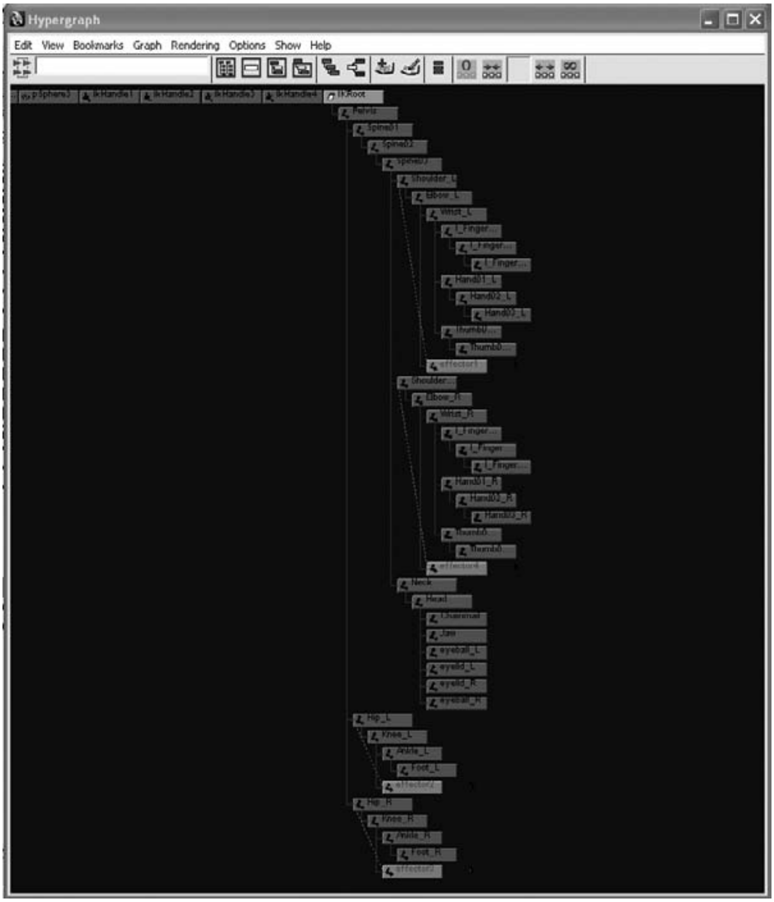


Figure 7.28
Add a group above the Pelvis joint.

You *could* parent the IK handles to the Pelvis joint, but I have found that it is often advantageous to have them separate when animating. Instead, create a group above the Pelvis joint and attach them there. Here's how:

1. Select the Pelvis joint and press Ctrl+G to create a new group, as shown in Figure 7.28. Name the group IKRoot.

2. Select the four IK handles. Then, while holding the Shift key down, click the IKRoot group and parent the IK handles to it as shown in Figure 7.29.

The skeleton is now finished and ready to attach to the model. Make sure you save your work!

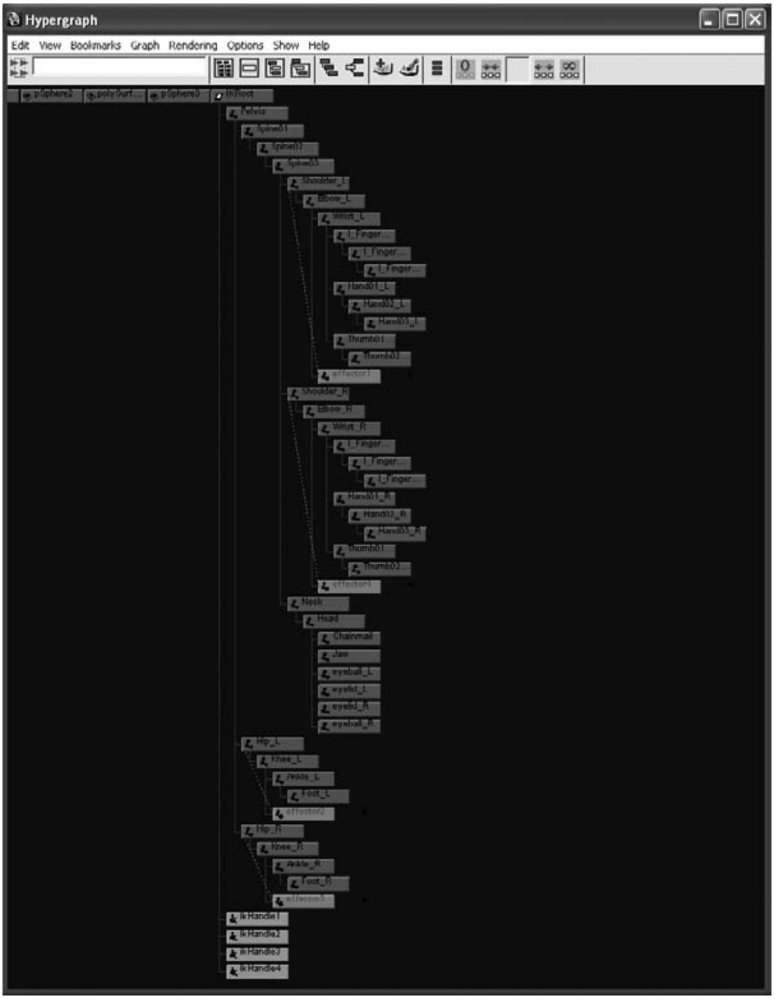


Figure 7.29
Make the four IK handles the direct children of the IKRoot group.

Skinning the Model

In Maya, the term used for attaching a skeletal system to a model is *skinning*. When a model is skinned, every vertex in the model is assigned to one or more joints. If a vertex is assigned to a single joint, it is called *rigid binding*. Rigid binding is good for mechanical objects like a robotic arm but doesn't work well for organic characters. If a vertex is assigned to more than one joint, it is called *smooth binding*. The first 3D characters in games were almost always rigid-bound models. As computing power increased, it became possible for one vertex be attached to several joints. Today, most characters are smooth bound. Smooth binding increases the natural look of an animation because the vertices are averaged between joints in some places.

When smooth binding is used, a vertex still share influences between multiple joints based on a percentage between 1 and 0. Therefore, if, for example, a vertex is assigned to both the shoulder joint and to the elbow joint but to no other joints, the influence of the two joints would equal 1. The higher the percentage for any joint, the more influence that joint has on the vertex.

Maya automatically calculates influence in a model and applies the influences during the skinning process. In most cases Maya does a pretty good job of attaching the skeleton to the model, but there are times when the artist needs to adjust the *skin weighting*—that is, the amount of influence a joint has on a vertex—of the vertices. Fortunately, Maya has some very powerful tools for adjusting skin weights.

To smooth-bind your model to the skeleton, do the following:

1. Select the model. Then, while holding down the Shift key, select the root joint of the skeleton (in this case, the Pelvis joint).

2. Open the Skin menu in the Animation main menu, choose Bind Skin, and select Smooth Bind to open the Smooth Bind Options dialog box, shown in Figure 7.30.

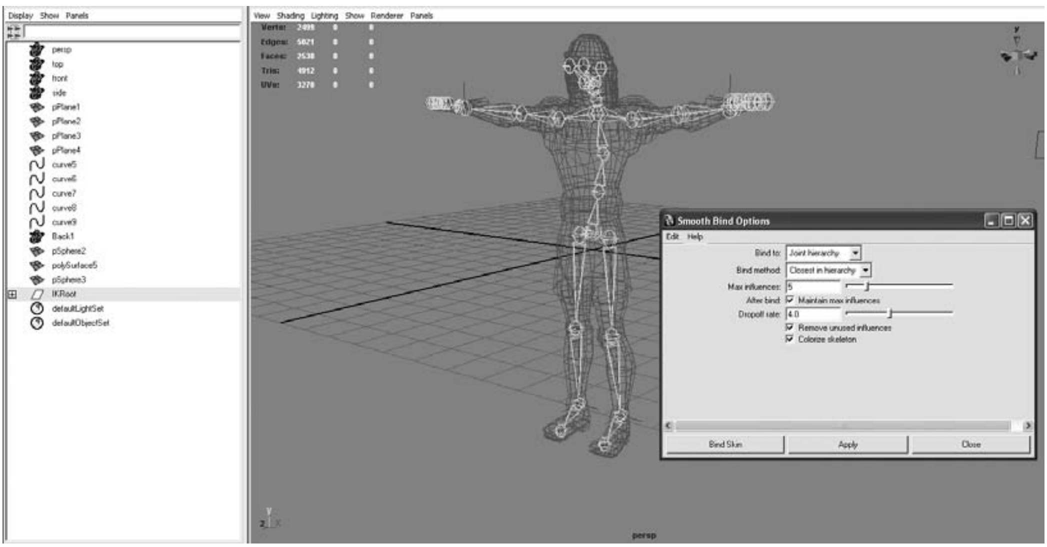


Figure 7.30
Soft bind the model to the skeleton.

3. From the Bind to drop-down list box, select Joint Hierarchy.
4. Set the Bind Method option to Closets Hierarchy.
5. In the Max Influence field, type 5.
6. Click the After Bind/Maintain Max Influence checkbox to select it.
7. Type 4 in the Dropoff Rate field.
8. Leave the Remove Unused Influences checkbox unchecked, but do click the Colorize Skeleton checkbox to select it.
9. Click the Apply button; the model is bound to the skeleton.

Because the eyeball joints are separate objects, you will need to bind them separately. Their binding method will be a little different however, because the eyeballs only need to be bound to a single joint. Do the following:

1. Select the eyeball geometry on the model. Then, while holding the Shift key down, click the eyeball joint to select it.
2. Open the Skin menu in the Animation main menu, choose Bind Skin, and select Smooth Bind to open the Smooth Bind Options dialog box (see Figure 7.31).

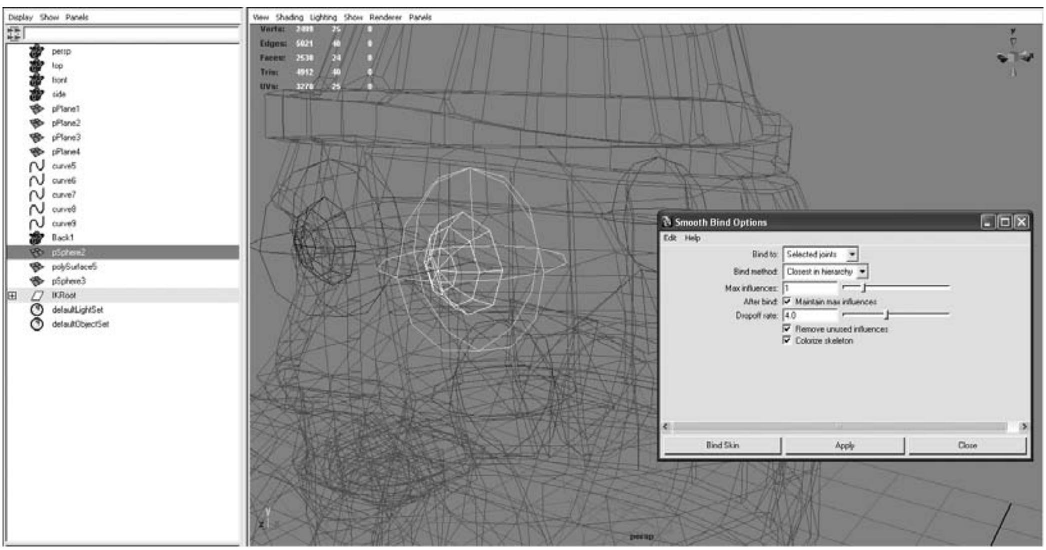


Figure 7.31
Use these settings to soft-bind the eye joints to the model.

3. From the Bind to drop-down list box, select Selected Joint.
4. Set the Bind Method option to Closets Joint.
5. In the Max Influence field, type 1.
6. Click the After Bind/Maintain Max Influence checkbox to select it.
7. Type 4 in the Dropoff Rate field.
8. Leave the Remove Unused Influences checkbox unchecked, but do click the Colorize Skeleton checkbox to select it.
9. Click the Apply button; the eye geometry is bound to the eye joint.
10. Repeat the binding process for the other eye. Your model is now bound to the skeleton!

Try moving a joint; notice how the model follows. Try rotating one of the joints to see how it works. (After you rotate the joint press, Ctrl+Z to put the joint back in its bound position.)

Adjusting Skin Weights

The binding process is pretty good, but it does present a few problems. To see what I mean, grab the left arm IK handle and pull the arm down as shown in Figure 7.32. Notice how the side of the chest caves in? That's because the Shoulder_L joint is influencing part of the chest. This and a number of other parts of the model need to have their skin weighting adjusted. (Press Ctrl+Z to put the arm back in place.)

Maya's Paint Skin Weights tool offers a very intuitive way to adjust skin weights. It allows the artist to apply skin weighting as if he were painting the model. Here's how it works:

1. Press the 5 or 6 key to choose flat or smooth shading respectively, and select the model.
2. In the Animation main menu, open the Skin menu, choose Edit Smooth Skin, and select Paint Skin Weights to open tool shown in Figure 7.33.
3. The model turns black and gray, with the black areas indicating those parts of the model that are not being influenced by the joint and lighter areas

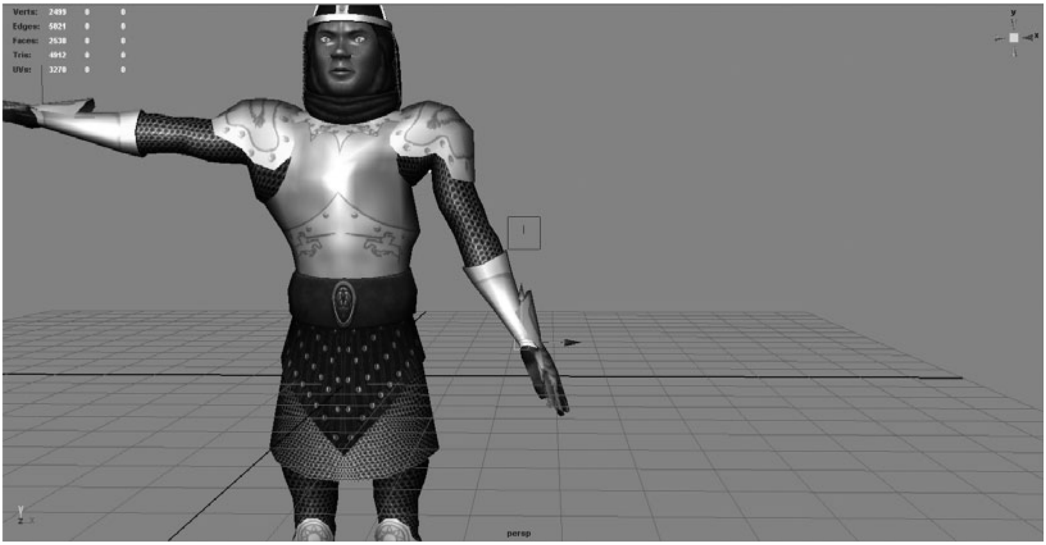


Figure 7.32
The current binding has problems.

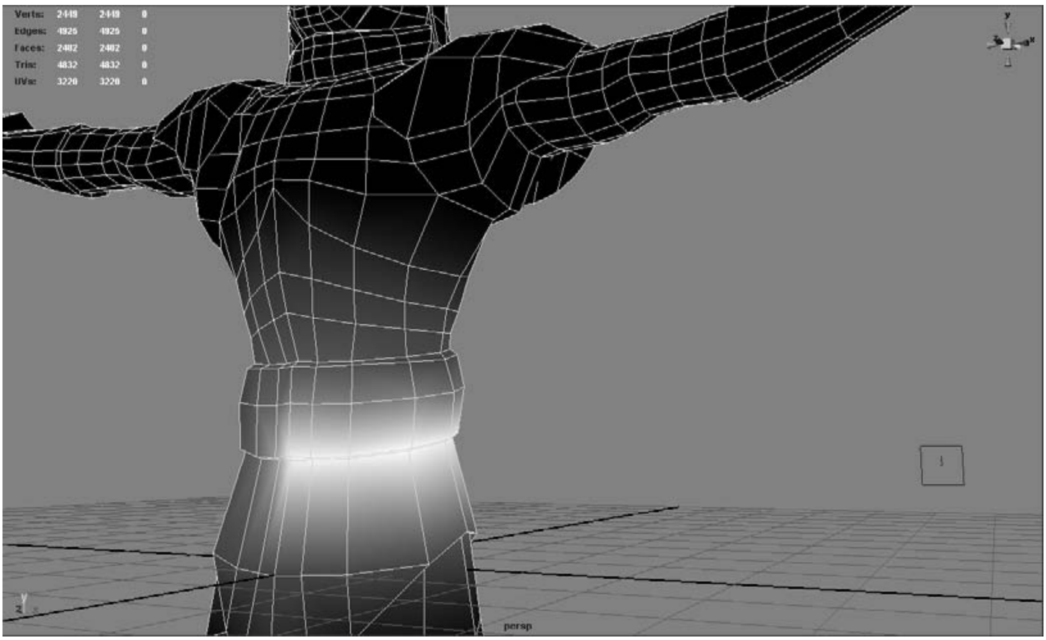


Figure 7.33
Launch the Paint Skin Weights.

indicating parts of the model being influenced by the joint. Notice, too, that there is a list of joints on the right side of the tool's dialog box. Click any joint in the list to see which vertices that joint influences.

4. When a joint is selected in the list, you can paint influence for that joint on the model. Try it by first select the Pelvis joint from the list and, using the following settings, paint along the bottom of the belt:
   - Radius (U): .2
   - Radius (L): .001
   - Opacity: .25

As you paint along the bottom of the belt, notice how the area that you're painting gets lighter. You can adjust the opacity of the brush to add more or less influence with each stroke.

Although the Paint Skin Weights enables you to quickly and easily fine-tune the skin weights of a model, some areas of the model call for a more direct approach. For these areas, use the Component Editor. With it, you can select individual vertices and change their individual values. For example, suppose the character's wrist guard is attached to his gauntlet and moves with the hand, but needs to be bound only to the joint at the wrist. To make that happen, you'd follow these steps:

1. In the Animation main menu, open the Windows menu and choose General Editors.
2. Click the Smooth Skins tab in the Component Editor to adjust the skin weights.
3. Select all the vertices in the left wrist guard. Then, in the Wrist_L column of the Component Editor, change the weighting for the wrist joint influence to 1. This ensures that all the vertices are attached to only the one joint, as shown in Figure 7.34.

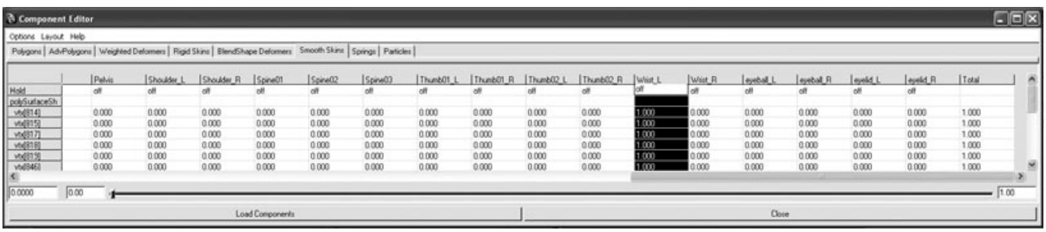


Figure 7.34
Change the binding of the wrist guard so it's bound to the Wrist_L joint only.

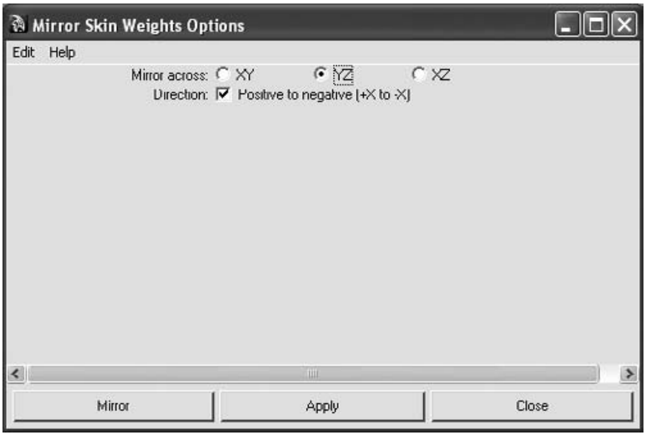


Figure 7.35
You can mirror the skin weights from this dialog box.

4. To mirror these smooth skin weights on the right side of the model, open the Skin menu in the Animation main menu, choose Edit Smooth Skin, and select Mirror Skin Weights to open the Mirror Skin Weights Option dialog box, shown in Figure 7.35.

Adjusting the skin weights is a time-consuming process that takes some experimentation to get just right. I usually adjust the weights to what I *think* will work and then test it by rotating one or more joints to see how things go. In this way, I can get a good weighting for each joint. The following is meant to help you determine which weights to use for various joints in your model, using the model we've been working on as an example:

- **Pelvis.** The pelvis shown in Figure 7.36 should be bound to the area just below the belt, with diminishing influence as you go down the skirting.
- **Spine01.** This joint is bound primarily to the area around the upper part of the belt, as shown in Figure 7.37.
- **Spine02.** Spine02 influences the lower half of the character's breastplate. Because it is made out of metal, this area is rigid; for this reason, the influences are at or near 1 for each vertex, as shown in Figure 7.38.
- **Spine03.** The top spine joint influences the top of the breastplate and the area that attaches the shoulder plate to the breastplate, as shown in Figure 7.39.

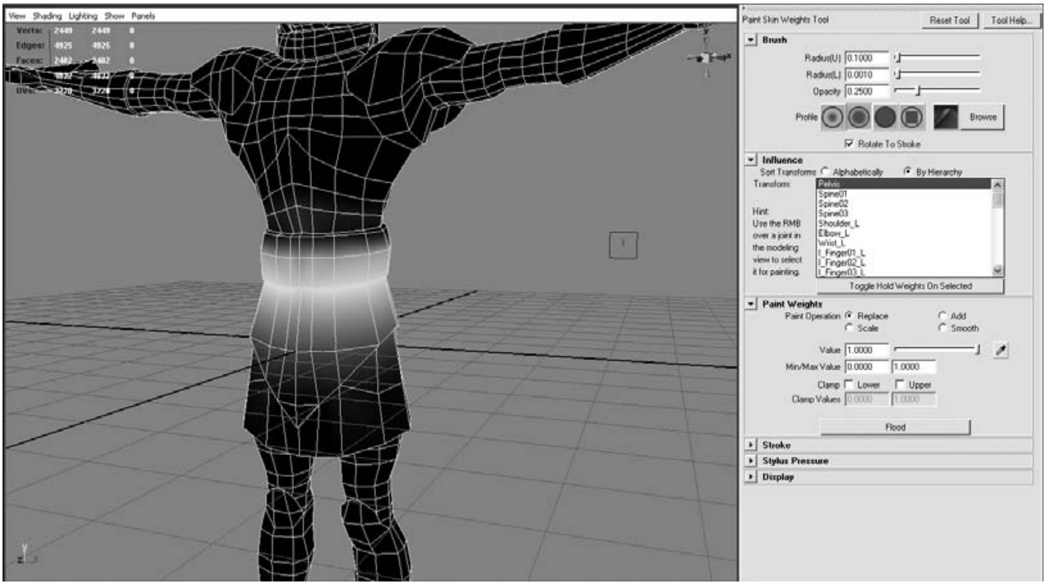


Figure 7.36
The pelvis affects mainly the area around the belt.

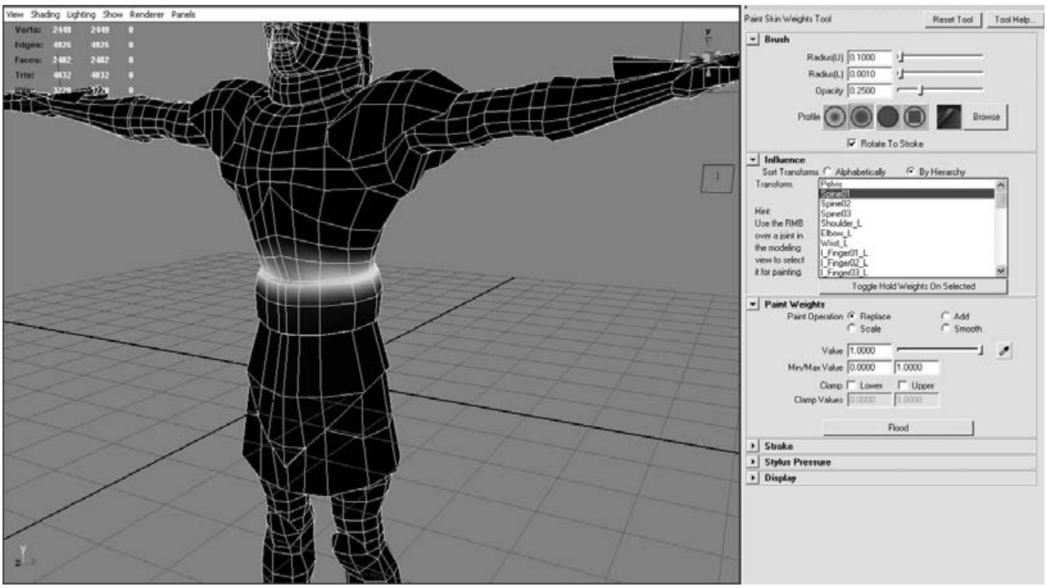


Figure 7.37
Spine01 influences the area around the top of the belt.

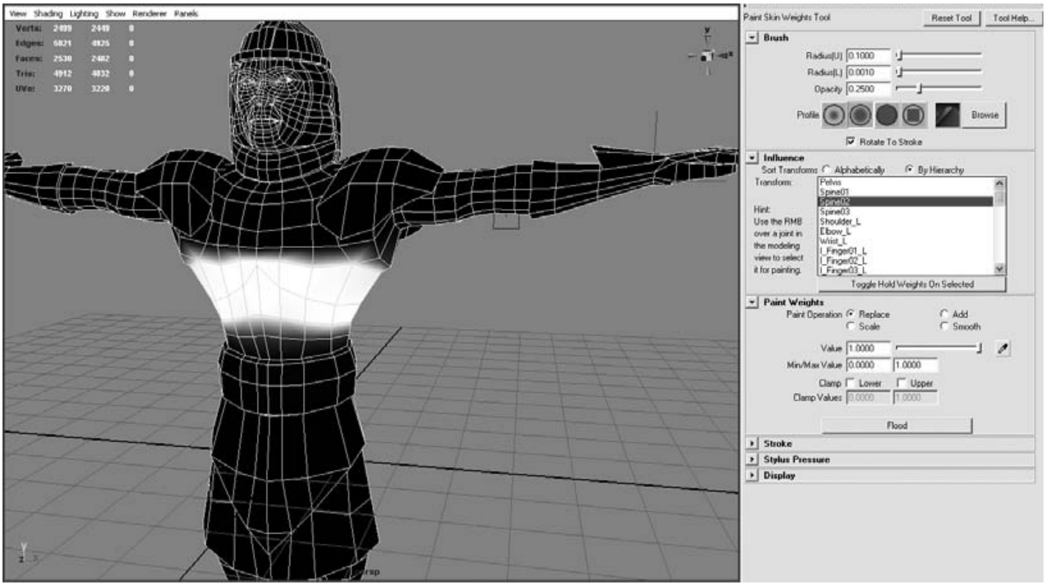


Figure 7.38
Spine02 influences the lower half of the breastplate.

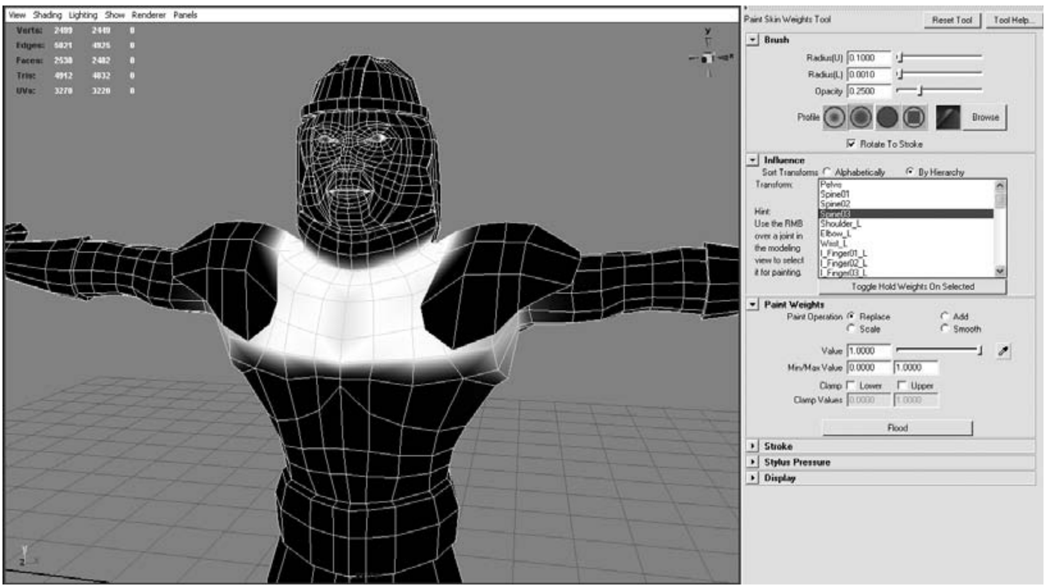


Figure 7.39
Spine02 influences the upper half of the breastplate.

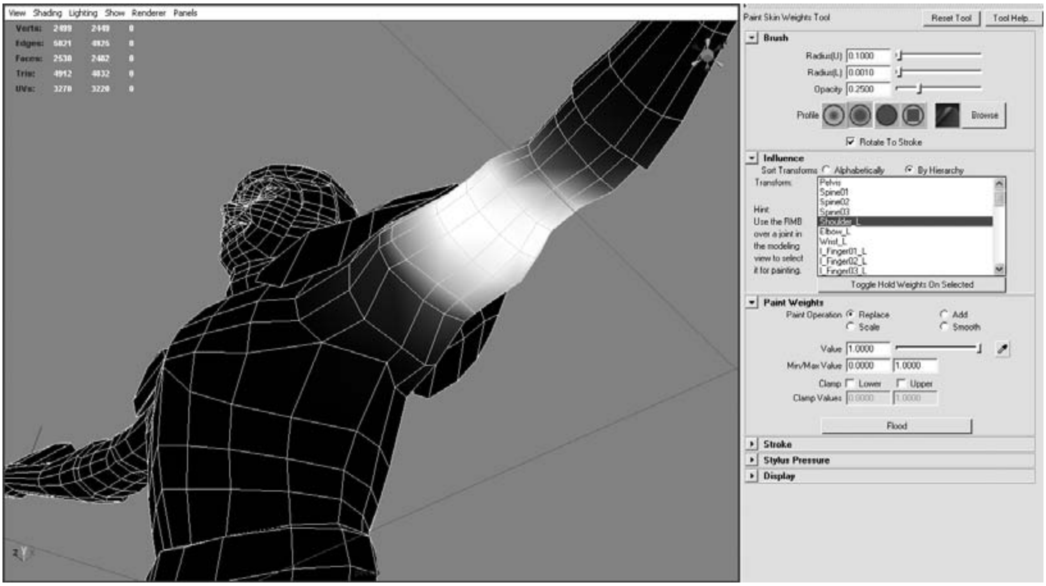


Figure 7.40
Shoulder_L influences the upper arm.

- **Shoulder_L.** The left shoulder joint influences the upper arm. When adjusting the weighting of this joint, you'll want to remember two important things: Add more influence to the area on the back of the arm around the elbow bone and less influence around the armpit, as shown in Figure 7.40.
- **Elbow_L.** The elbow joint influences the lower arm down to the wrist, as shown in Figure 7.41.
- **Wrist_L.** The wrist influences the palm and back of the hand as well as the wrist guard, as shown in Figure 7.42.
- **I_Finger01_L.** The first index finger joint influences the first knuckle of the finger, as shown in Figure 7.43.
- **I_Finger02_L.** The next finger joint influences only the second knuckle of the index finger, as shown in Figure 7.44.
- **I_Finger03_L.** The last finger joint influences the tip of the finger, as shown in Figure 7.45.
- **Neck.** The Neck joint influences only the area around the character's neck wrap, as shown in Figure 7.46.

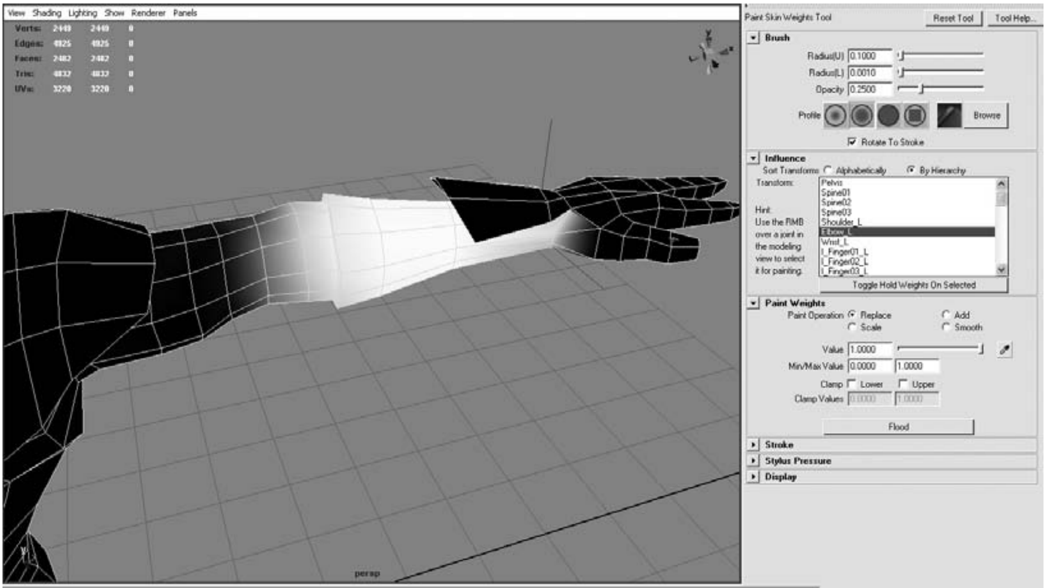


Figure 7.41
The elbow influences the lower arm to the wrist.

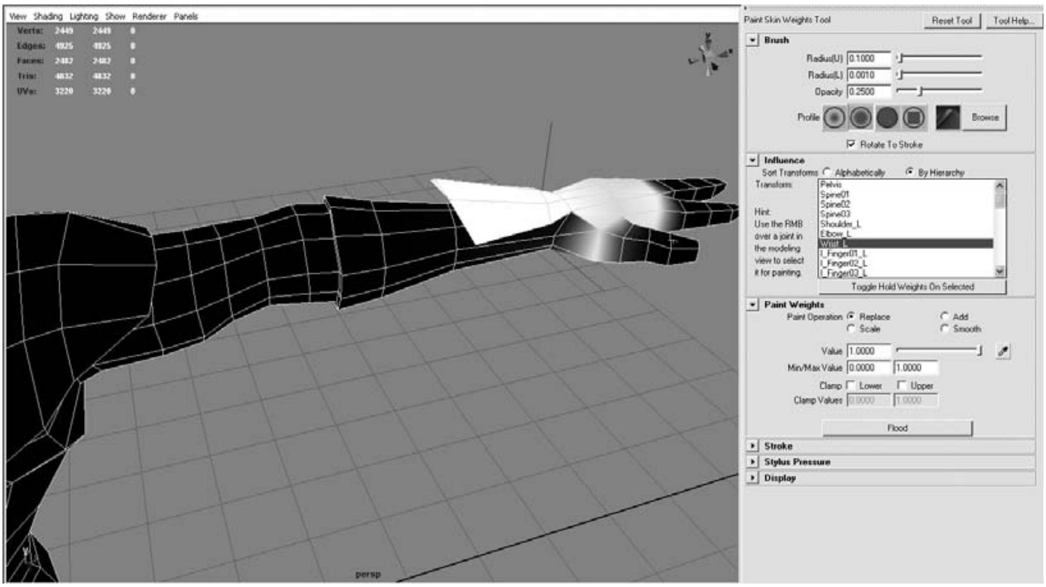


Figure 7.42
The wrist joint influences the hand and wrist guard.

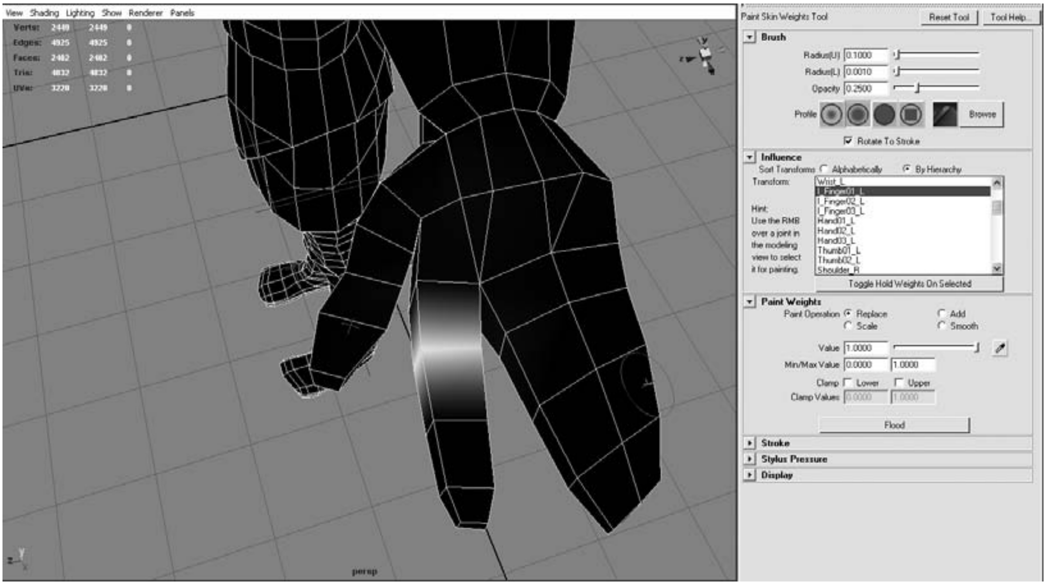


Figure 7.43
The first finger joint influences only the first knuckle of the index finger.

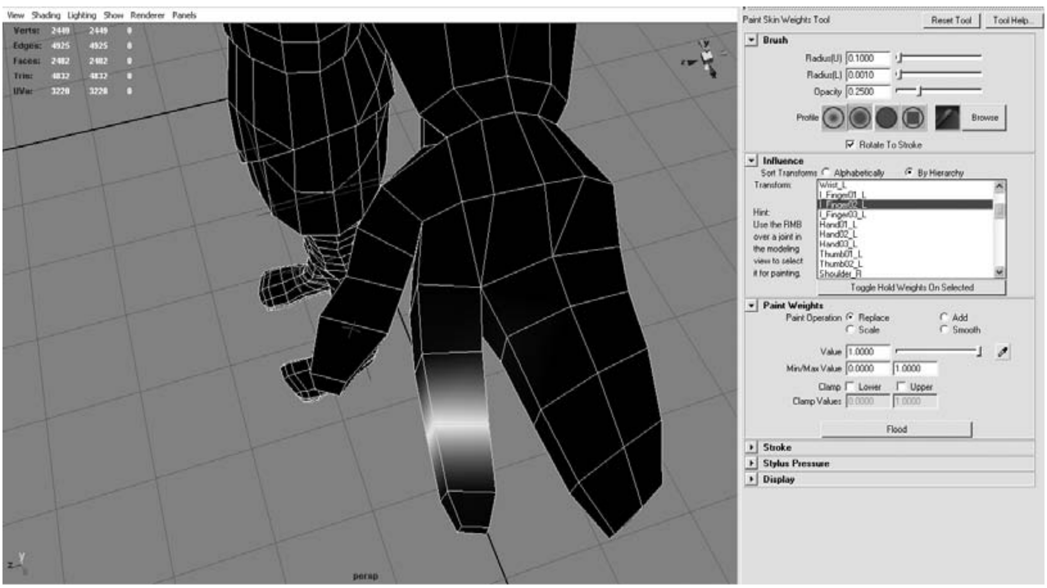


Figure 7.44
The second finger joint influences the second knuckle.

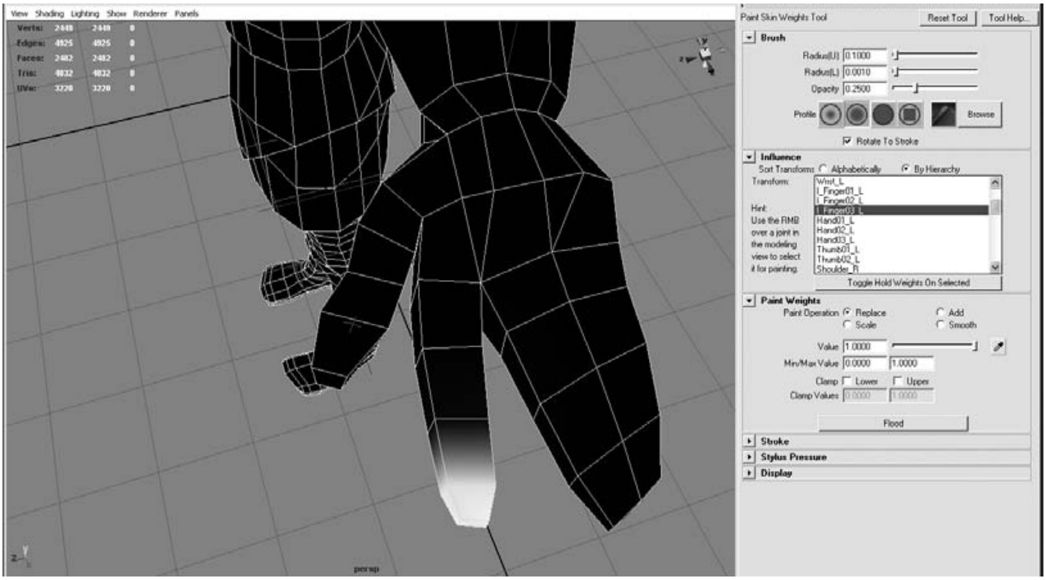


Figure 7.45
The last finger joint influences the tip of the finger.

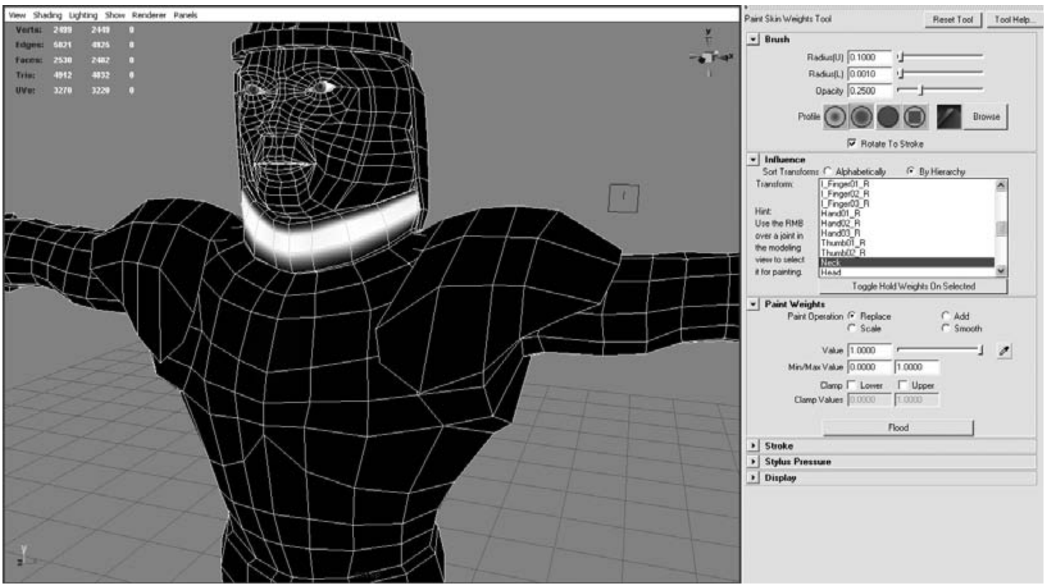


Figure 7.46
The influence of the neck joint is contained in the area around the neck.

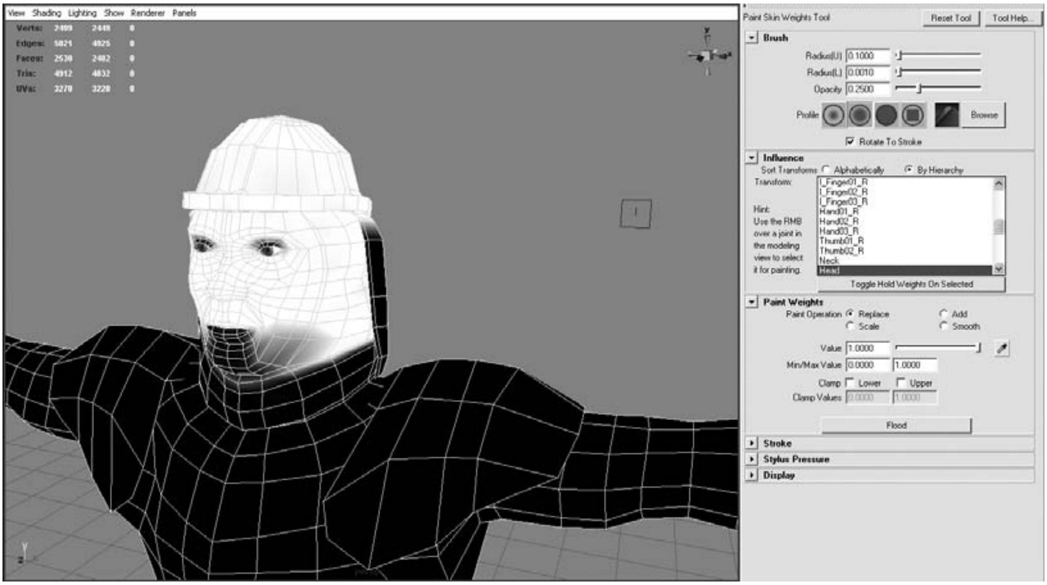


Figure 7.47
The head joint influences most of the head.

- **Head.** The head is a fairly rigid part of the body, so most of the areas of the head except for around the eye and jaw are strongly bound to the Head joint, as shown in Figure 7.47.
- **Jaw.** The Jaw joint strongly influences the lower lip and chin, diminishing along the lower cheeks as shown in Figure 7.48. It also influences the lower half of the inner portion of the mouth, including the lower teeth.
- **Eyelid_L.** The Eyelid_L joint influences the area around the upper eyelid. You need to test this one to make sure it covers the eye when the character blinks, as shown in Figure 7.49.
- **Chainmail.** The Chainmail joint's influence is on the chainmail at the back of the head, diminishing toward the corners as shown in Figure 7.50. This is a little tricky because the chainmail has both an inside and an outside, and you have to get both sides. I used the Component Editor to adjust the vertices because I couldn't see the inside vertices to paint with the Paint Skin Weights tool.
- **Chainmail_L.** The right and left corners of the chainmail behind the head are influenced by the Chainmail_L joint, as shown in Figure 7.51, and by the Chainmail_R joint (not shown).

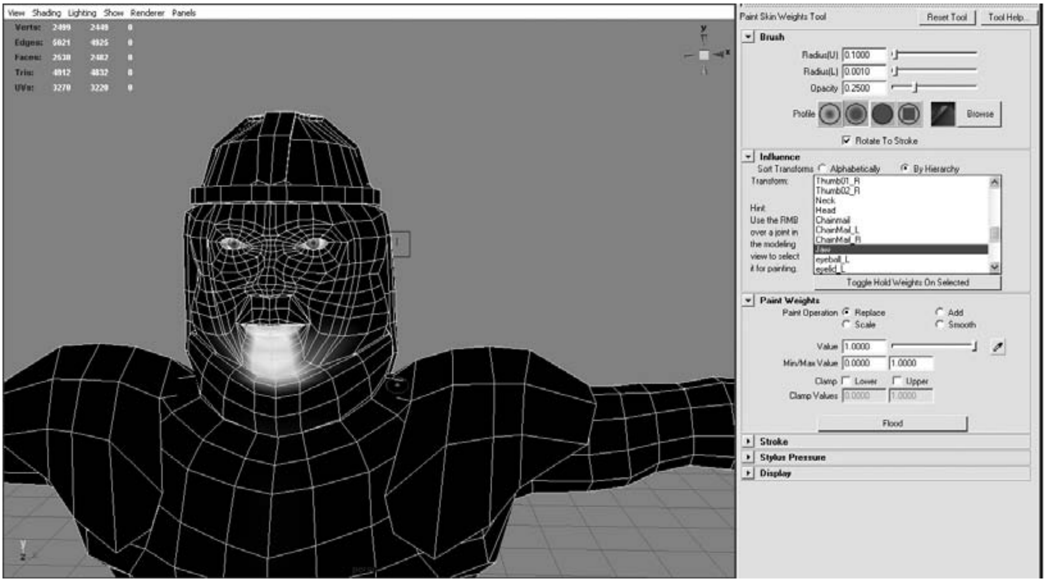


Figure 7.48
The jaw controls the chin and lower mouth.

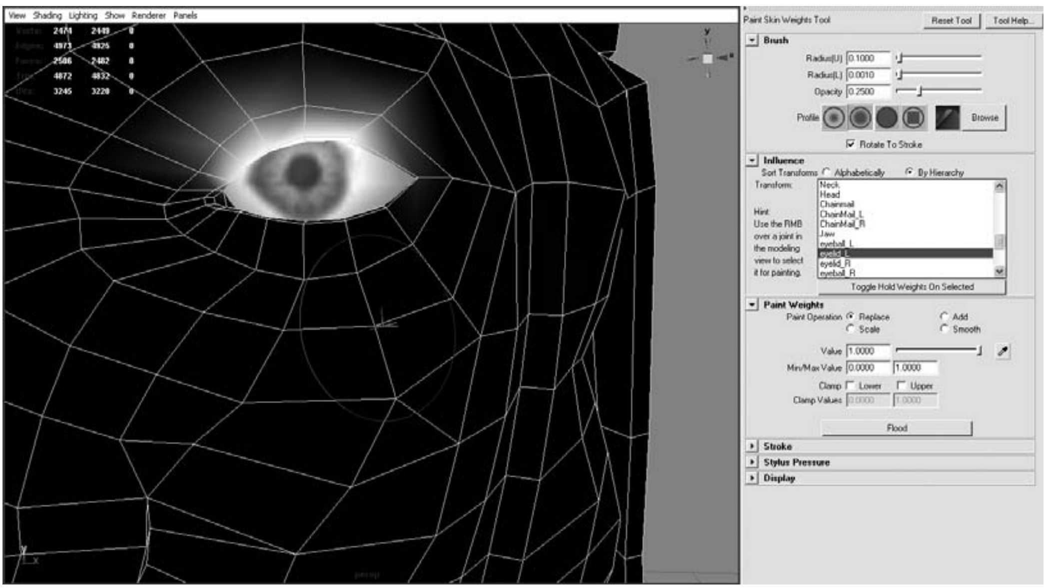


Figure 7.49
The Eyelid_L joint influences the area immediately above the eye.

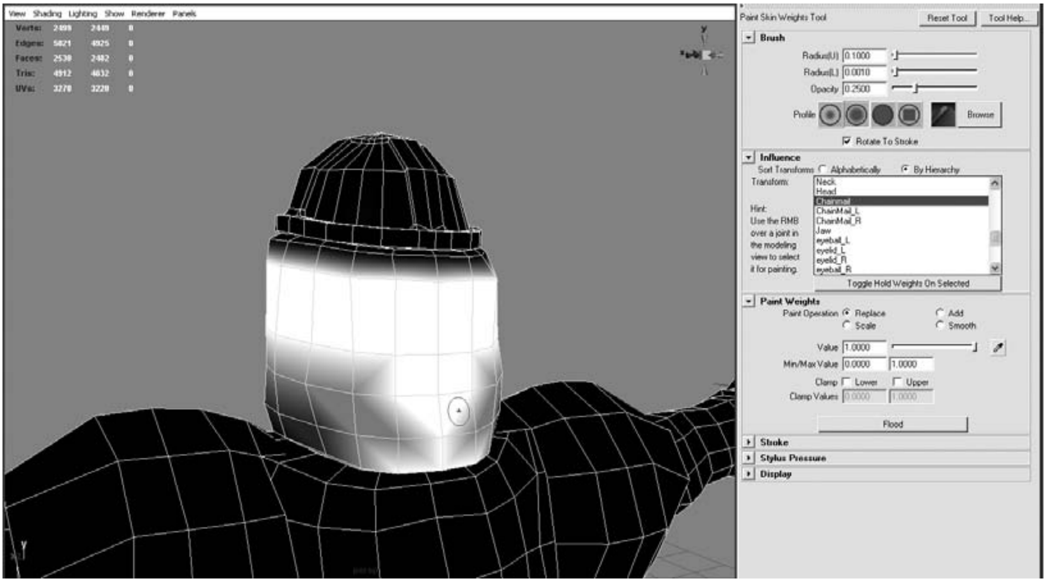


Figure 7.50
The Chainmail joint influences the chainmail at the back of the neck.

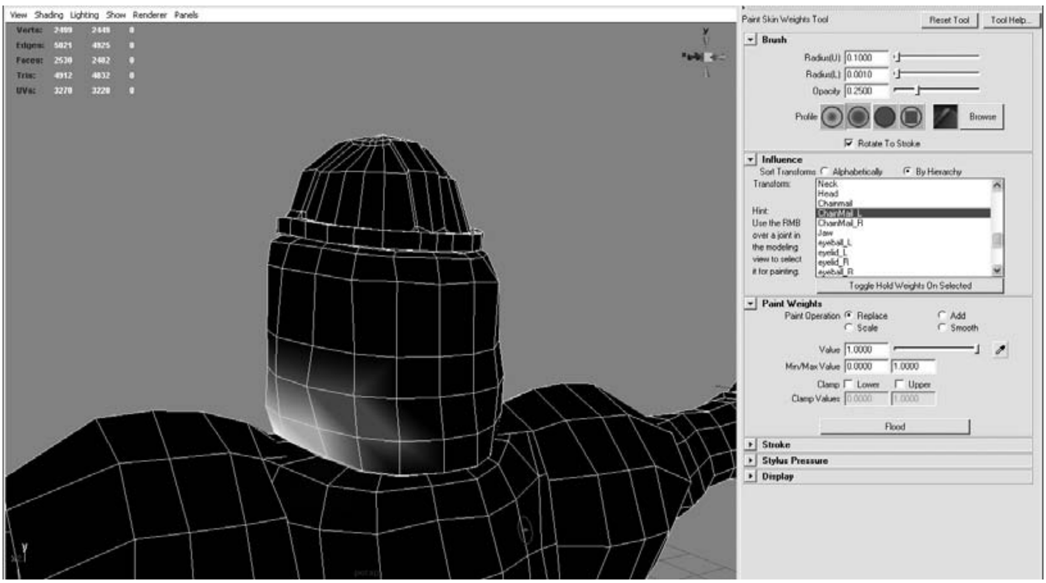


Figure 7.51
Chainmail_L influences the left side of the chainmail.

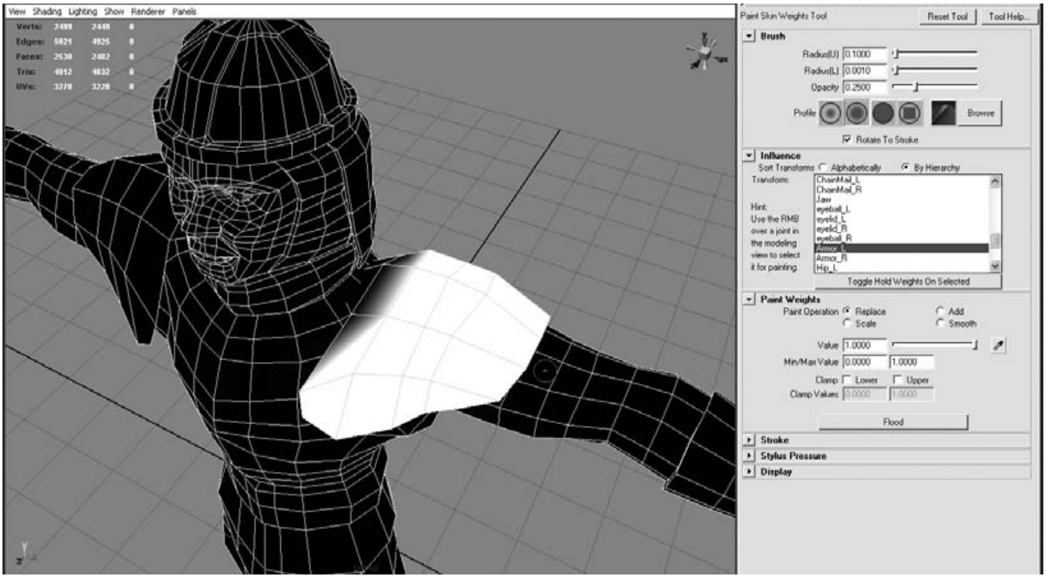


Figure 7.52
Armor_L influences the shoulder plate.

- **Armor_L.** Armor_L influences the shoulder plate almost exclusively—except for the area where it attaches to the breastplate, as shown in Figure 7.52. Like the chainmail, the geometry also has a top and bottom.
- **Hip_L.** The Hip_L joint influences the upper leg area, down to the knee. Additionally, both the Hip_L and Hip_R joints evenly influence the vertices on the line in the middle between the two hip joints, as shown in Figure 7.53.
- **Knee_L.** The Knee_L joint influences the lower leg and the greaves extending above the knee cap, as shown in Figure 7.54.
- **Ankle_L.** The ankle joints influence most of the foot, as shown in Figure 7.55.
- **Foot_L.** The foot joints influence the area from the ball of the foot to the tip of the toes, as shown in Figure 7.56.

Hopefully, these examples will help you as you work on the skin-weighting of your model. Like I said earlier, the best way to weight the model is to test it and change weights where needed. Keep working at it until you get it just right.

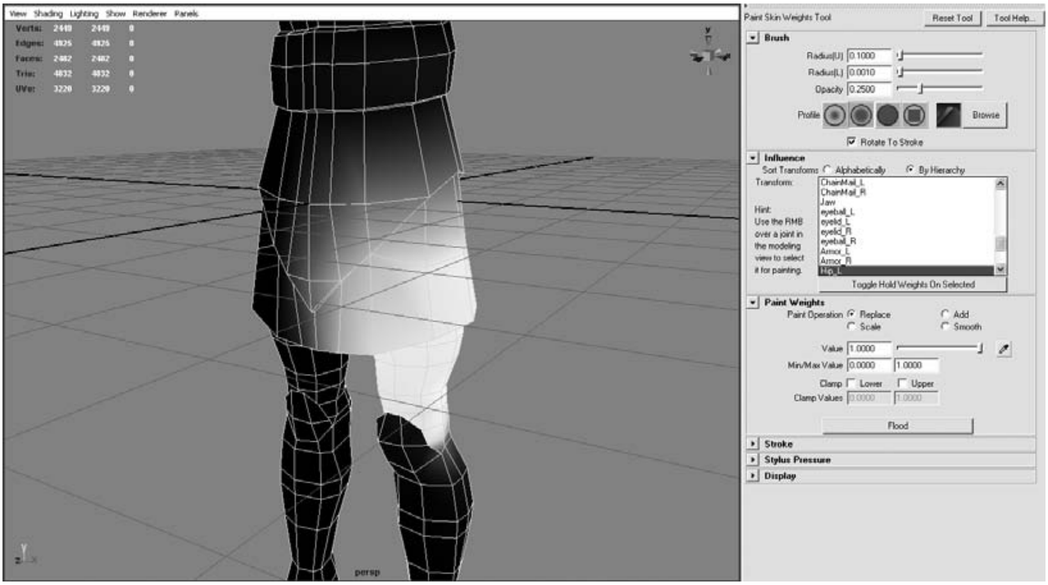


Figure 7.53
The hip joints influence the upper leg.

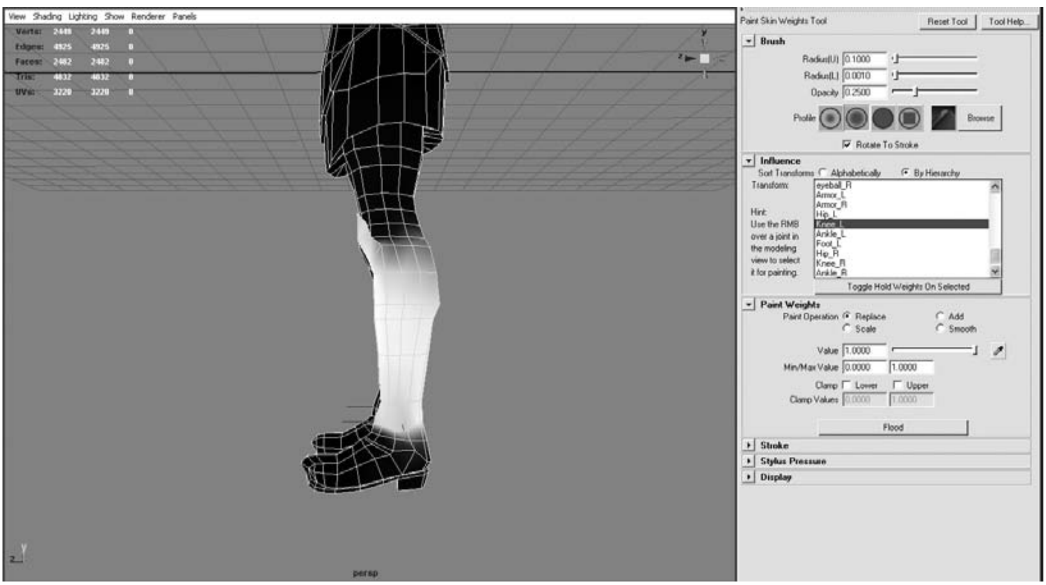


Figure 7.54
The knee joints influence the lower leg and greaves.

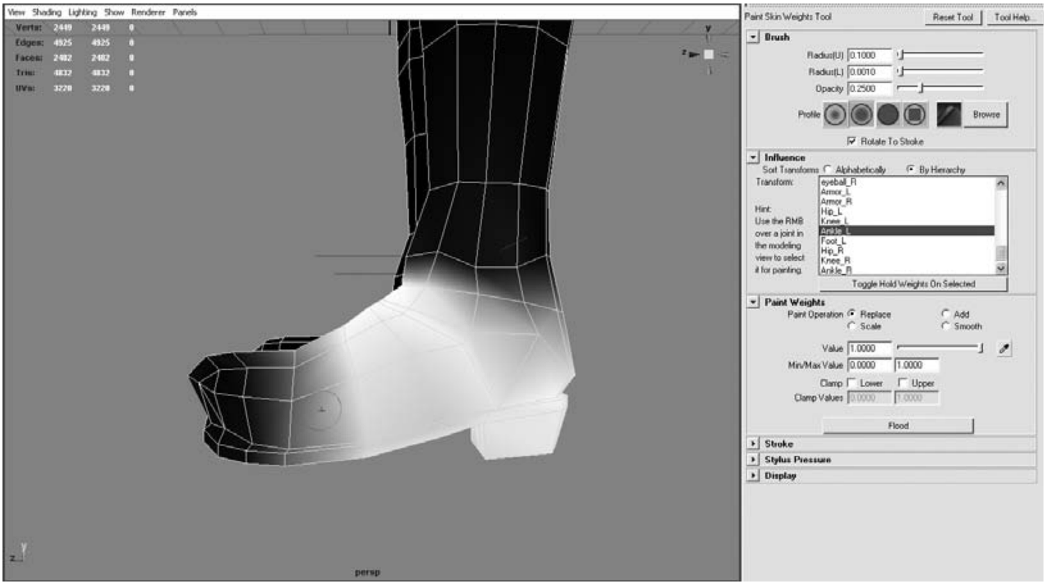


Figure 7.55
The ankle joints influence the feet.

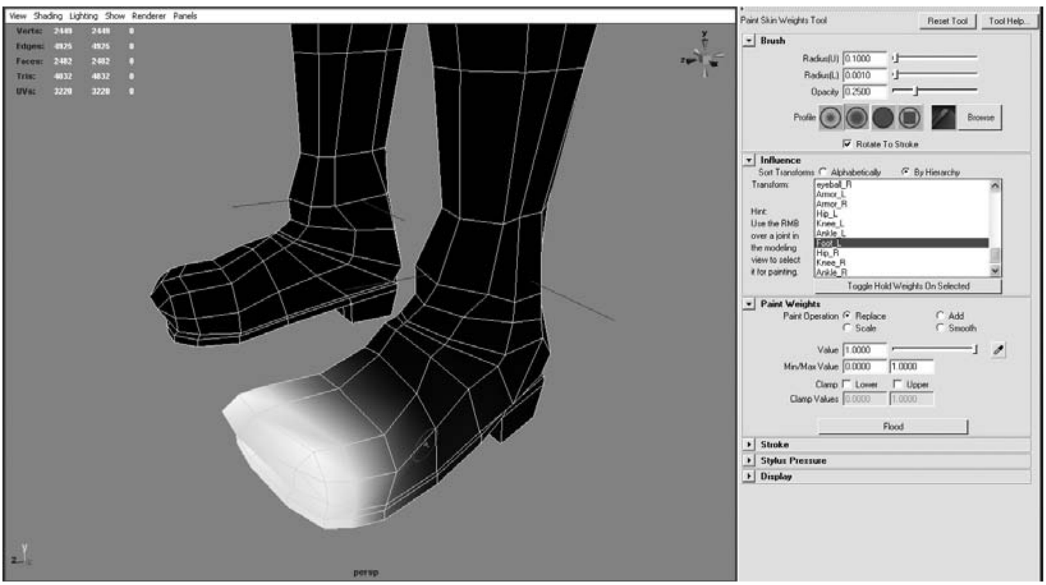


Figure 7.56
The Foot_L joint influences the toes of the left foot.

Summary

This chapter covered a lot of ground with regards to rigging a character for animation. That said, there is a lot to learn beyond the material covered here. If you followed the examples, however, and rigged your character model, you should now have a good understanding of the basics.

In this chapter you learned about the following:

- Skeletal systems for character animation
- Building joints and bones
- Placing joints in a model
- Connecting joints in a hierarchy
- Mirroring joints across a model
- Creating IK handles
- Smooth and rigid binding
- Applying smooth binding to a character
- Adjusting skin weights using the Paint Skin Weights tool
- Adjusting skin weights using the Component Editor
- Mirroring skin weights

CHAPTER 8

Basic Animation

Animation is the process of displaying a series of pictures that successively show motion in rapid succession, with the slight differences between each picture giving the illusion of movement. Although many people think of animation as related specifically to drawn images such as cartoons, the term actually refers to all forms of moving pictures, including motion pictures, videos, television programs, and video games. After all, a motion-picture camera does not record movement per se; rather, it records a series of still images called *frames*, each frame a single image in a series.

In motion pictures and television, these frames are presented so quickly—normally 24 frames per second in motion pictures (although the frames per second, or *fps*, can be as high as 70) and 30 fps in television and video—that the human eye does not register them as individual images. In contrast, frame rates in video games are not set, and are instead determined by the speed of the computer and the complexity of the program. That said, most game developers target 60 fps. Games that drop below 30 fps are harder to play because the controls often seem sluggish. To help you get a handle on animating your own game characters, this chapter covers a lot of basic animation concepts, presented in the form of individual projects in 2D and 3D animation to help you learn them in theory and practice.

2D Animation

When I first started creating games, all animation was done in 2D, or what is sometimes referred to as *cell animation*, a term that comes from the motion-picture industry. It refers to how animation for cartoons was once drawn on clear plastic pages called *cells*, which were laid over background art and photographed one at a time. Likewise, although the art in these older games was created using pixels rather than cells, it, too, involved "laying" the action over background art, with each movement drawn by hand. This type of animation, whether for motion pictures or video games, was a very time-consuming process that required several artists working on a single project.

2D animation in games differed in two major ways from animation to motion pictures. First, the 2D game animator was not concerned with the flow of animation over a long period of time transitioning from one movement to another. That is, in games, the player usually chose an action, so the game animator just focused on individual actions, trying to make it so they all started and ended in nearly the same way so they could fit together. The second aspect of 2D game animation that was different from motion-picture animation was that the game animator didn't worry about camera transitions. The camera was locked such that the player's view was always from the same angle and distance.

Creating a Simple 2D Animation

Almost every full-featured paint program has features that allow you to create animations. This example uses Corel Painter because it has some unique features to help compare frames with previous frames.

1. You set up an animation in Corel Painter the same way you would set up a new picture: by opening the File menu and choosing New (see Figure 8.1).
2. In the New dialog box, choose the Movie option button in the Picture Type area, change the frame count to 16, and change the image size to 512 × 512 (see Figure 8.2). When you're finished, click OK.
3. The Enter Movie Name dialog box, shown in Figure 8.3, opens automatically. Name the animation Ball and click Save.
4. The New Frame Stack dialog box opens automatically; it allows you to set up Painter's Onion Skin feature, which simulates the vellum used by traditional animators to view multiple drawings stacked on top of each other. Choose 4

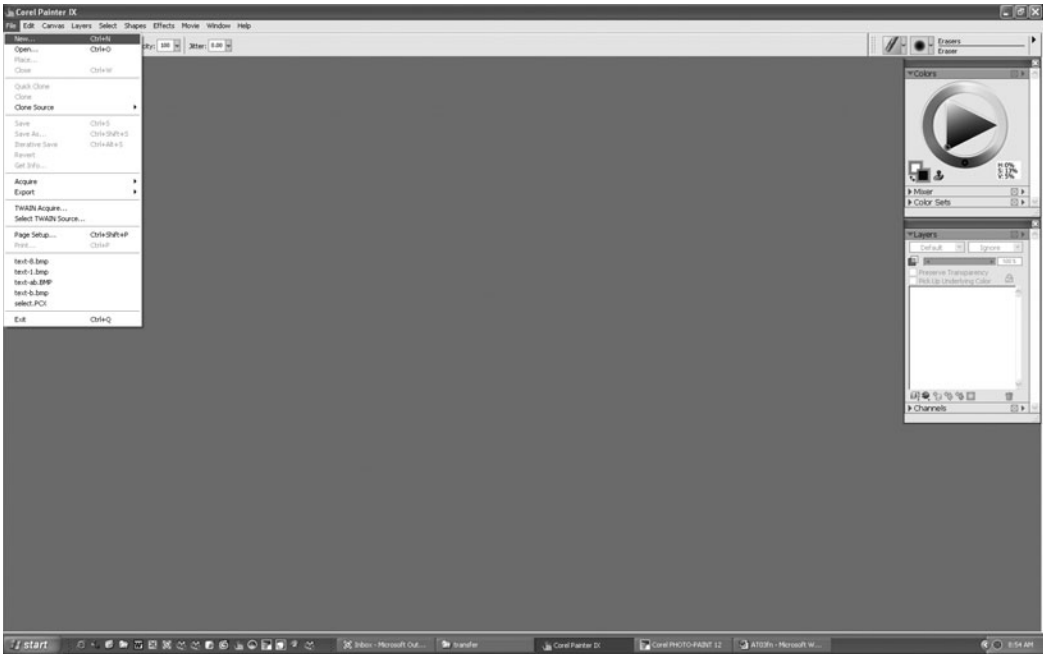


Figure 8.1
Select New from the File menu to set up a new animation.

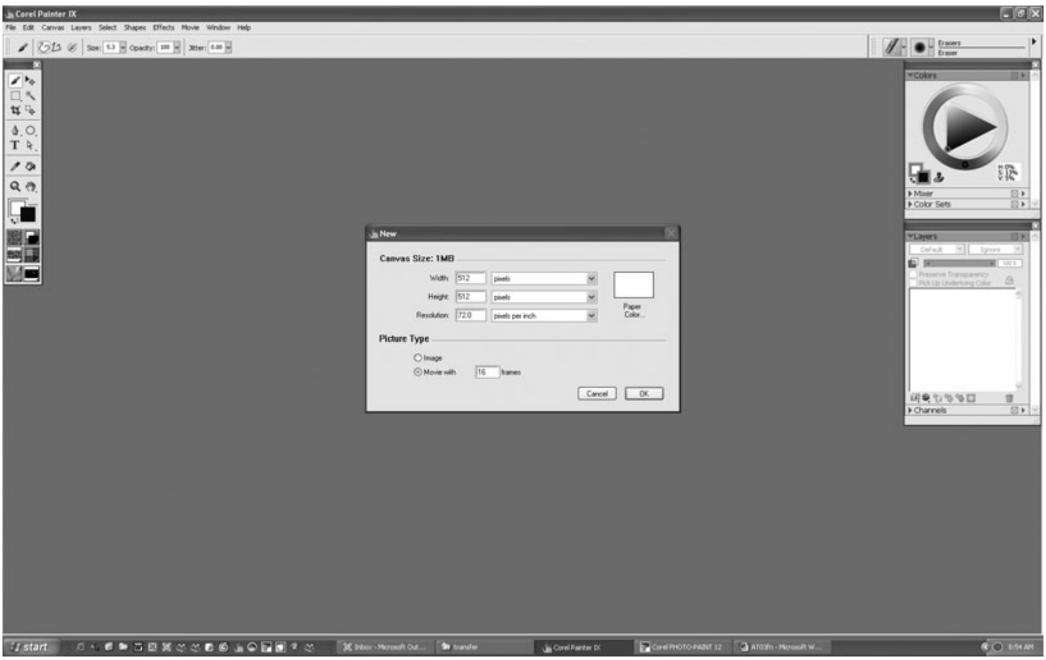


Figure 8.2
The New dialog box, set to create an animation

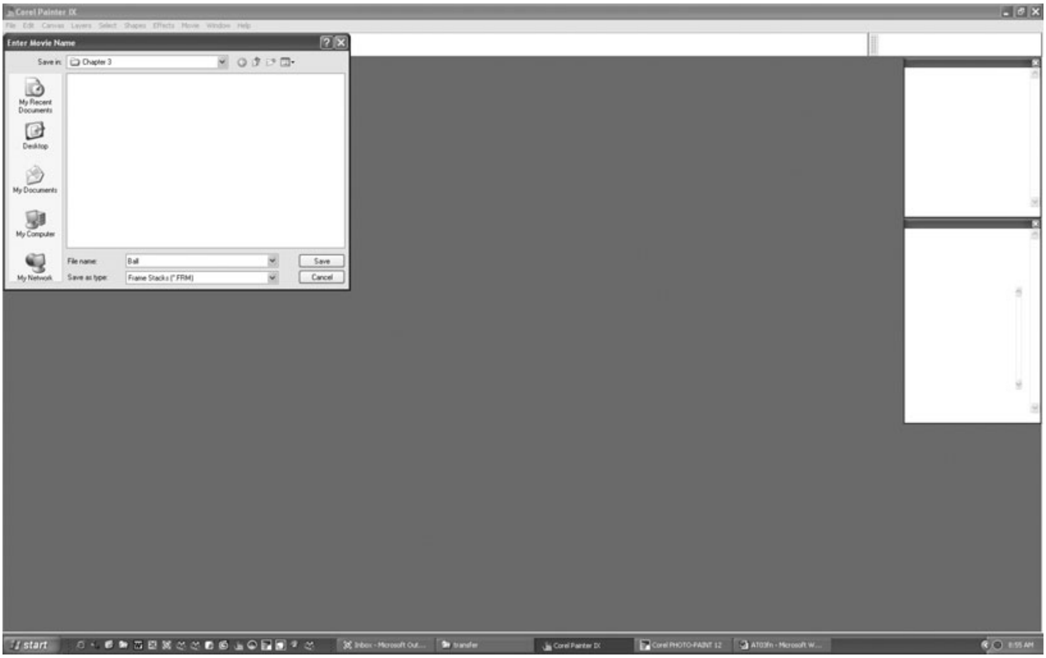


Figure 8.3
The Enter Movie Name dialog box

under Layers of Onion Skin and choose 24-bit color with 8-bit alpha under Storage Type, as shown in Figure 8.4. Click OK to continue.

5. The Animation and Frame Stack windows open automatically. In Corel, *frame stack* refers to the series of frames in an animation; the Frame Stack window contains controls for advancing frames forward and backward. The current frame is on the right, with the red arrow above it. Your first step is to add an image to the frame to animate. For this example, you'll animate an image of a marble such that it drops from the left side of the screen and bounces to the right. (Moving an item is the simplest form of animation because all you are doing is changing the position of the item from frame to frame. Later, you will work on moving a character, which is a bit more complicated.) Rather than drawing your own marble, use the one provided among Corel Painter's preset images; to begin, open the Window menu and select Show Image Portfolio, as shown in Figure 8.5. This opens the Image Portfolio.

6. Drag the marble image from the Image Portfolio to the Animation window (see Figure 8.6).

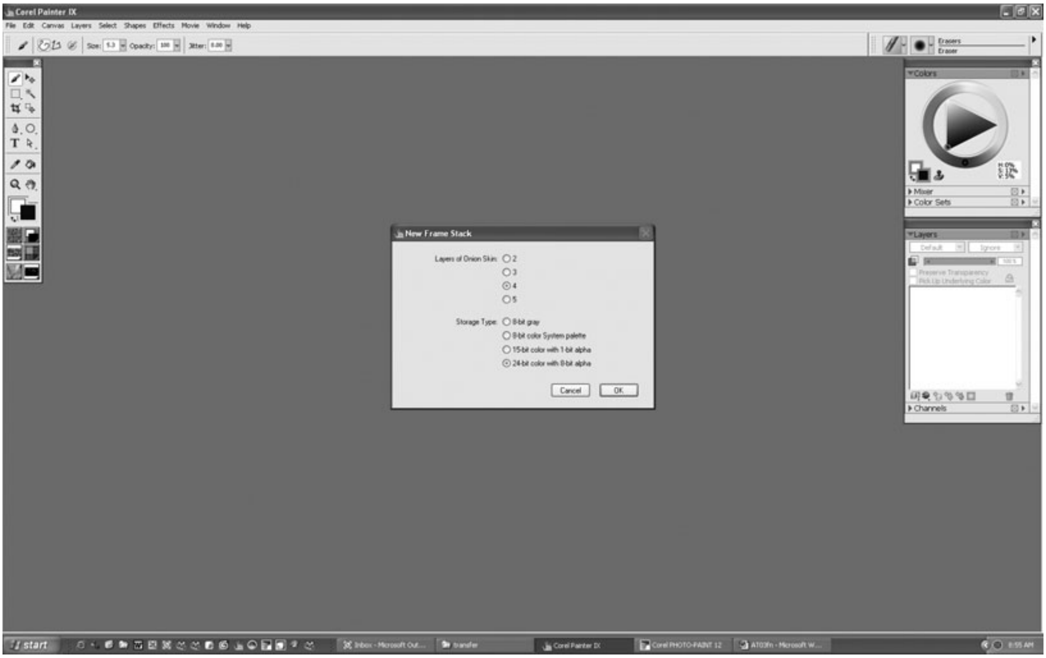


Figure 8.4
The Onion Skin level is set to 4.

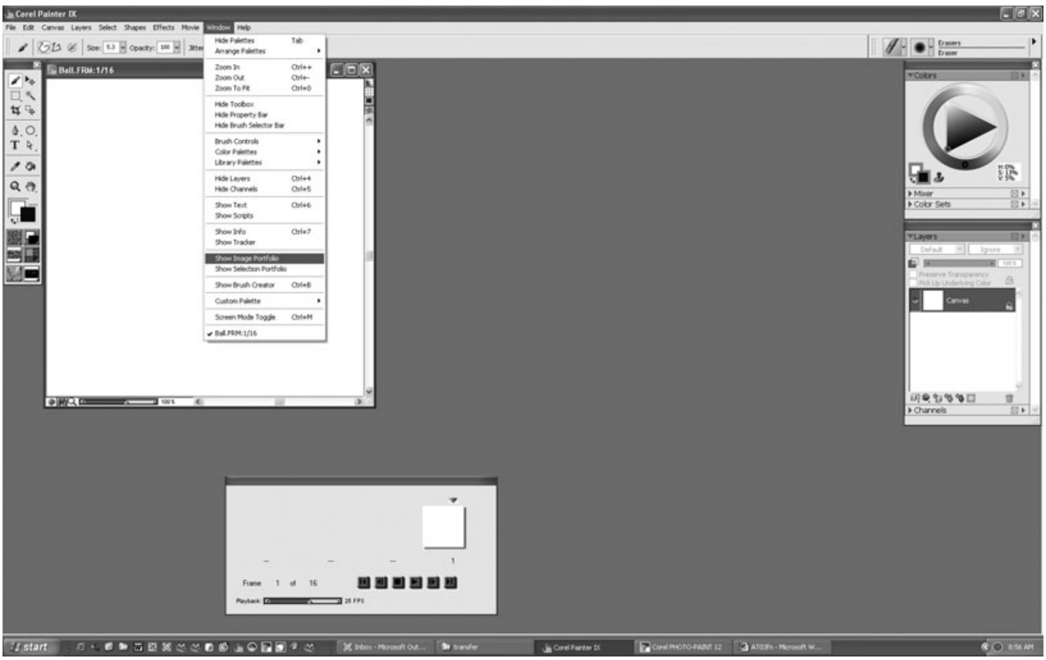


Figure 8.5
Select Show Image Portfolio from the Window menu.

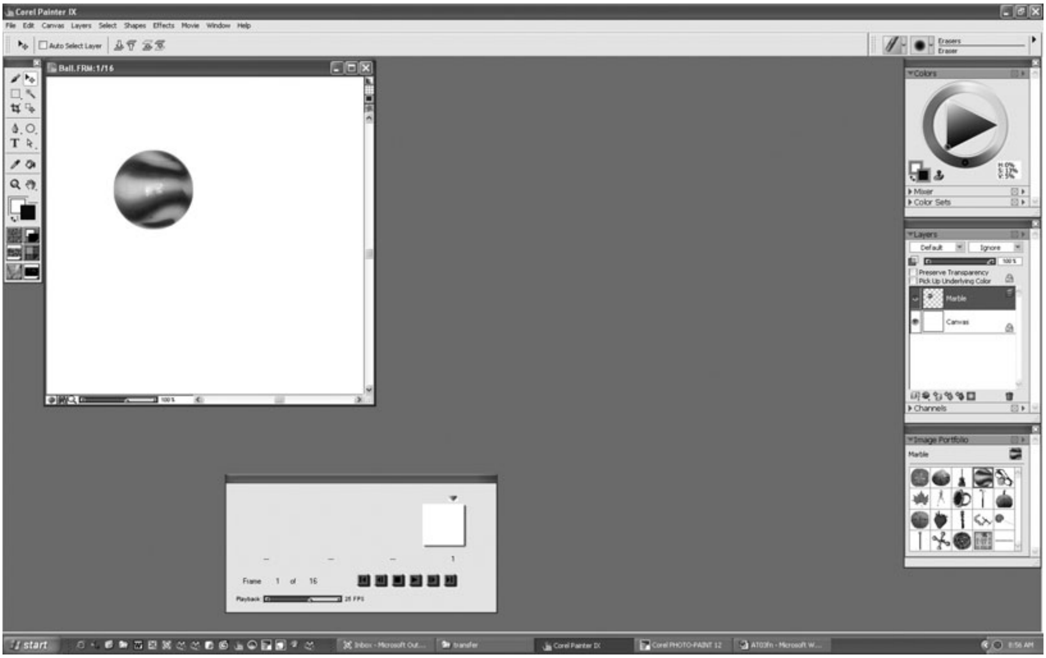


Figure 8.6
Move the marble image to the animation area.

7. Position the marble as high in the upper-left corner as possible (see Figure 8.7).
8. To use the Onion Skin feature, open the Canvas menu and choose Tracing Paper, as shown in Figure 8.8.
9. Click the Advance One Frame button in the Frame Stack window. (It's the second button from the right.) Notice that the marble is currently in the same position in frame 2 as in frame 1.
10. Drag the marble in frame 2 down and to the right, as shown in Figure 8.9.
11. For the sake of example, this animation will show the marble hitting the ground, represented by the bottom of the image, in frame 5. To mimic the way falling objects speed up as they descend, increase the spacing of the marble in frames 3 and 4, dragging it a bit farther down and to the right in each succeeding frame (see Figure 8.10). Notice how the placement of the marble in preceding frames is also visible, as though you are looking at them through tracing paper.

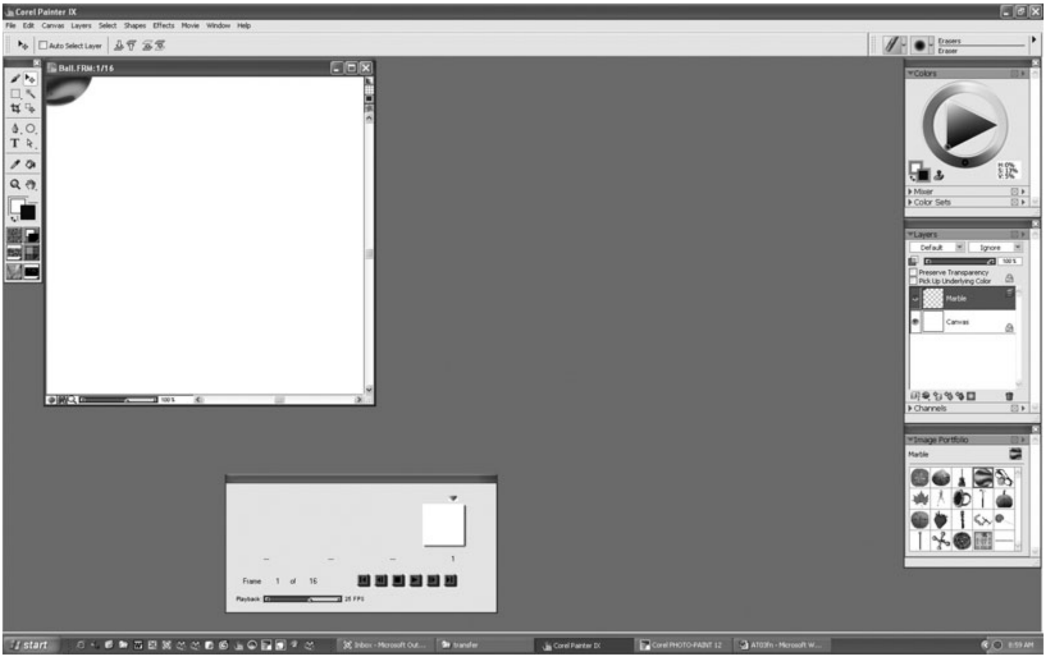


Figure 8.7
Position the marble in the top-left corner of the screen.

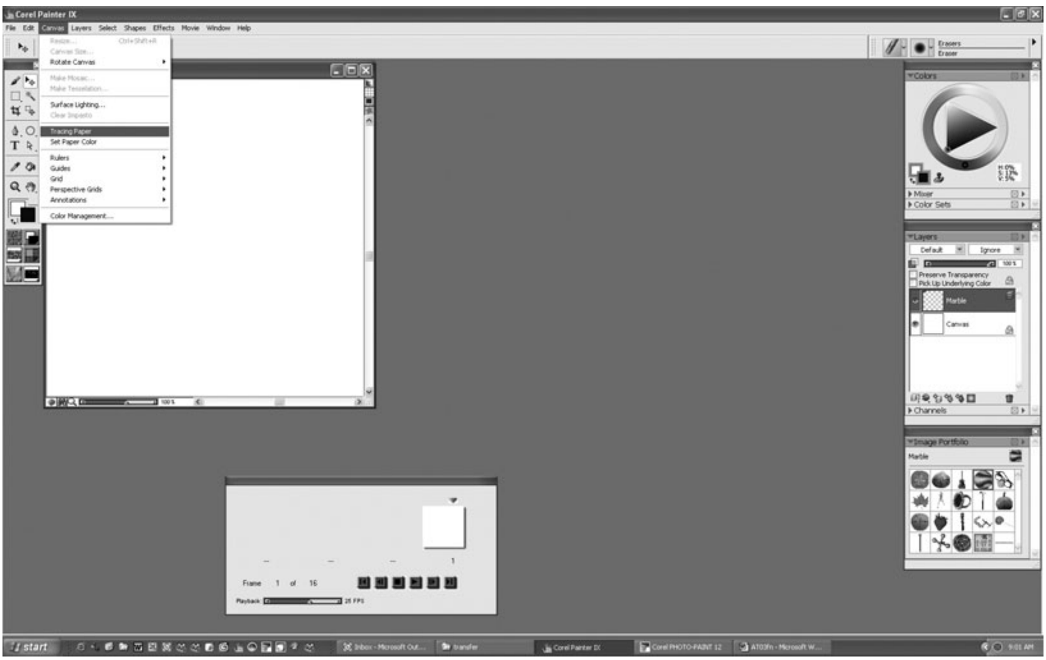


Figure 8.8
Select Tracing Paper to turn on the Onion Skin feature.

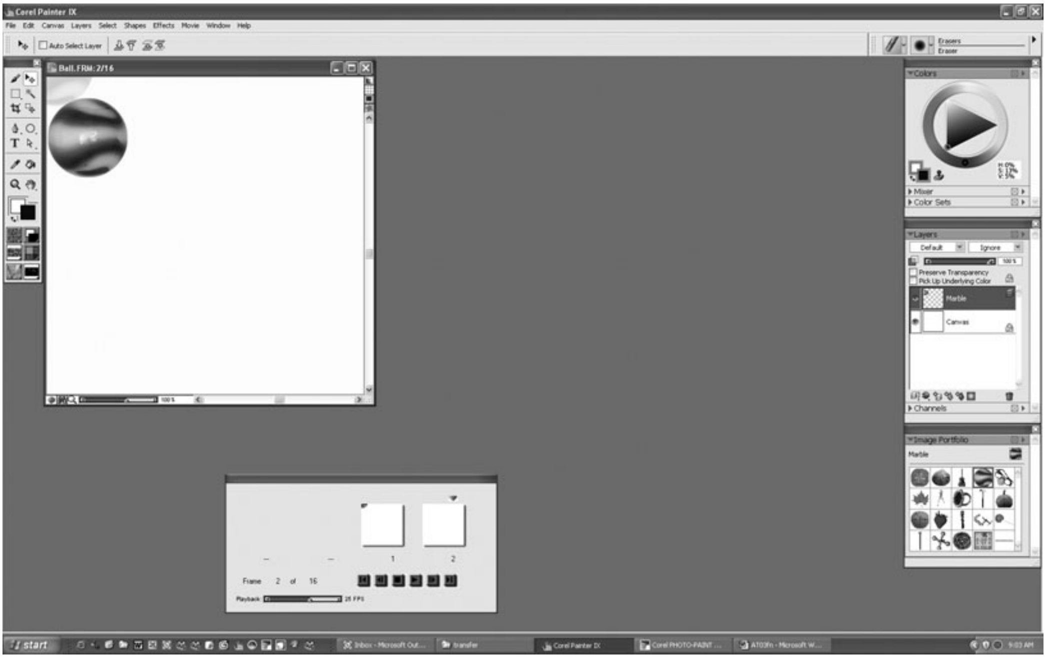


Figure 8.9
The marble is moved to a new position in frame 2.

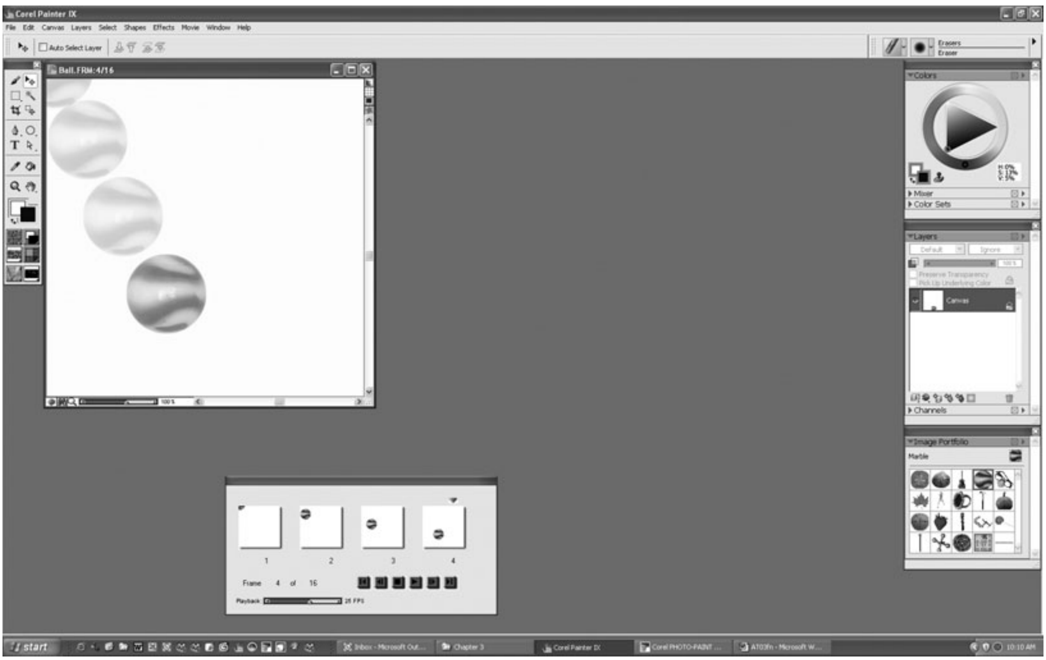


Figure 8.10
Separate the marble a little more in each succeeding frame.

Note

The most common mistake beginning animators make is setting an even difference in movement between frames. That is, they fail to take into account the fact that faster movements have greater differences between frames, while slower movements have lesser differences between frames. In animation, everything is a combination of time and distance. These two factors, as they relate to each other, are called *timing* in animation terms.

12. As the marble hits the ground, some of the energy from the falling marble is transferred to the ground, with the rest redirected in the bounce of the marble—the end result being that the marble slows down a bit. That means that in frames 5–10, the spacing marble of the marble should be decreased a bit. In addition, you'll want to squash the appearance of the marble slightly in frame 5 to indicate that it is hitting a hard surface. (Although this is only one frame, it is noticeable and will enhance the animation.) To begin, open the Effects menu, choose Orientation, and select Scale (see Figure 8.11).
13. In the Scale Selection dialog box, uncheck the Constrain Aspect Ratio checkbox, change the Horizontal Scale setting to 110.0 percent, and change

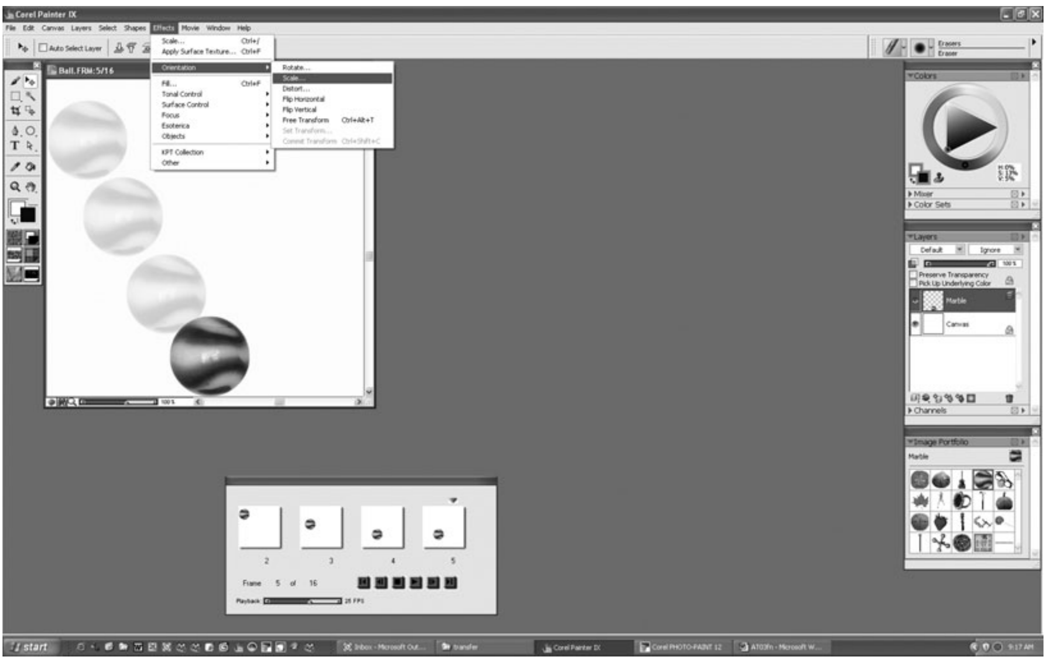


Figure 8.11
Select Scale.

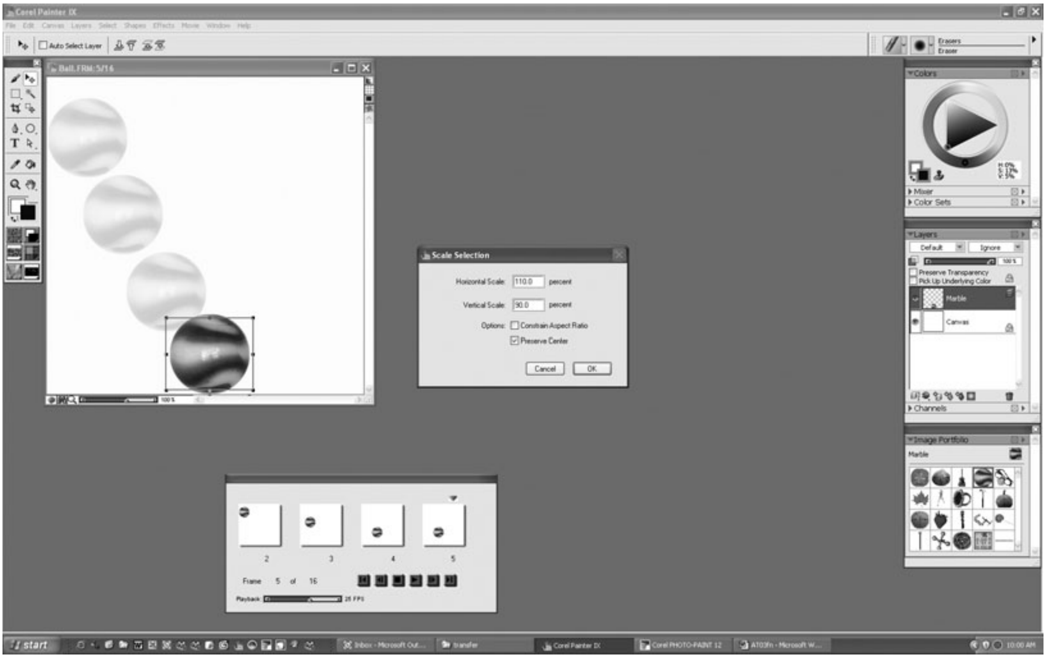


Figure 8.12
Change the scale of the marble.

the Vertical Scale setting to 90.0 percent (see Figure 8.12). Click OK. When you're finished, drop the layer by selecting Drop from the Layers menu. (That way, the distorted marble in this frame won't carry over to the next one.)

14. In frame 6, you want to stretch the marble slightly to indicate that it's springing from the ground. To begin, copy the marble from frame 1 and paste it into frame 6. Then open the Effects menu, choose Orientation, and select Rotate to open the Rotate Selection dialog box.

15. Rotate the marble 15 degrees, as shown in Figure 8.13.

16. Open the Effects menu, choose Orientation, and select Scale. Then, in the Scale Selection dialog box, scale the marble 90 percent horizontally and 110 percent vertically to get the stretch-y look. When you're finished, click OK.

17. Repeat steps 14 and 15, but this time rotate the marble −15 degrees to put it back where it was (see Figure 8.14). Again drop the layer so it won't carry over to the new frame.

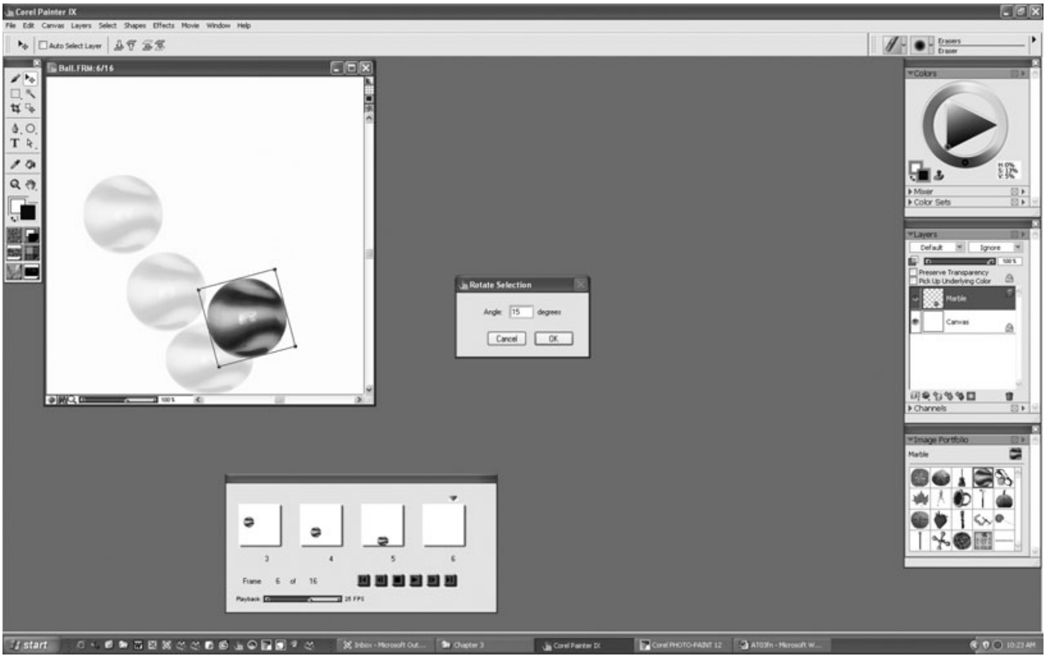


Figure 8.13
Rotate the marble 15 degrees.

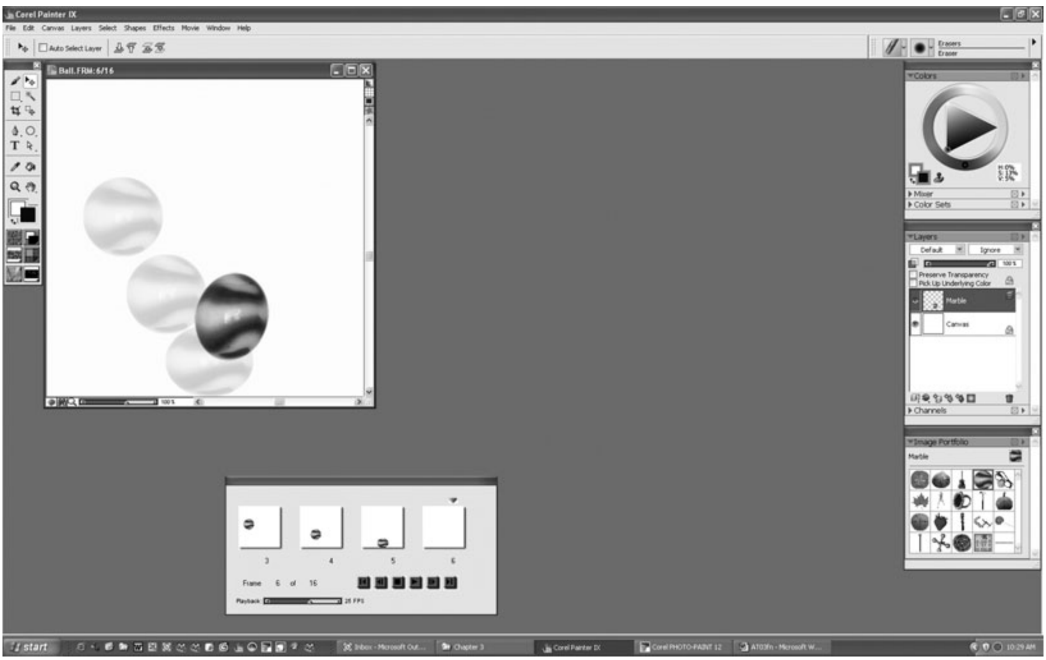


Figure 8.14
Stretch the marble in the direction of the bounce.

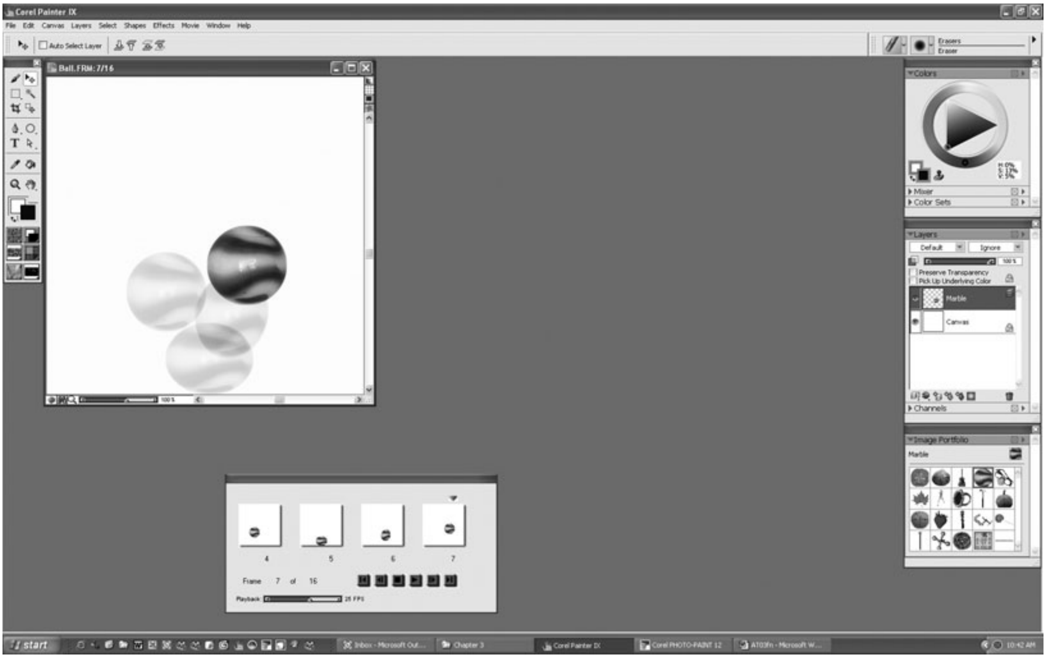


Figure 8.15
Place a new marble in frame 7.

18. Copy the marble from frame 1 and paste it into frame 7 placing it as shown in Figure 8.15.
19. Continue to move the marble up and to the right, decreasing the distance of movement slightly from frame to frame until you get to frame 12 (see Figure 8.16).
20. The marble will start to accelerate on its descent. Increase the distance moved by the marble between each succeeding frame as it descends off the frame (see Figure 8.17). Drop the marble layer at frame 16. Do not advance the animation to 17 because this is a 16 frame animation. Painter automatically creates new frames if you advance the frame.

Congratulations! You just finished an animation. Click the Play button (see Figure 8.18) to see how you did. You can set the frame rate using the slider bar.

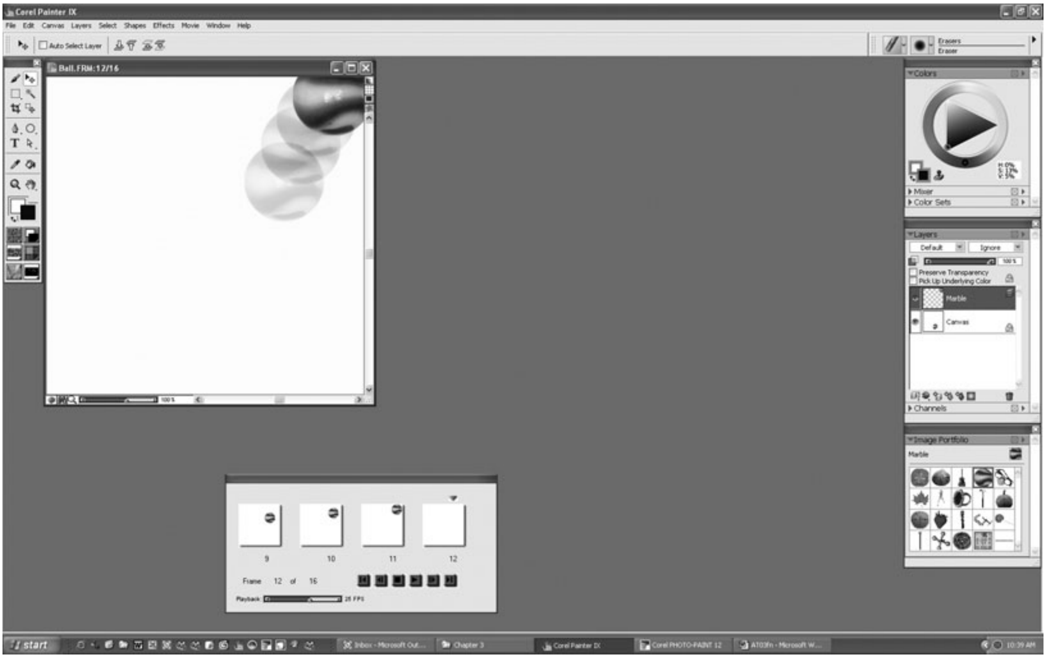


Figure 8.16
The marble slows as it reaches the top of its bounce.

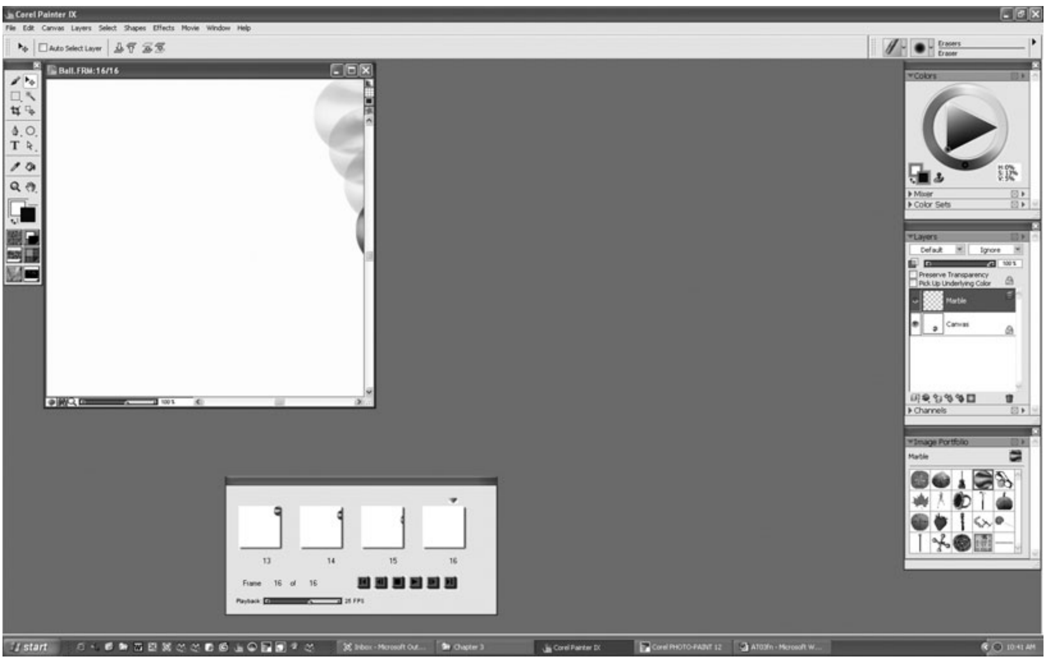


Figure 8.17
The marble will increase in speed as it descends.

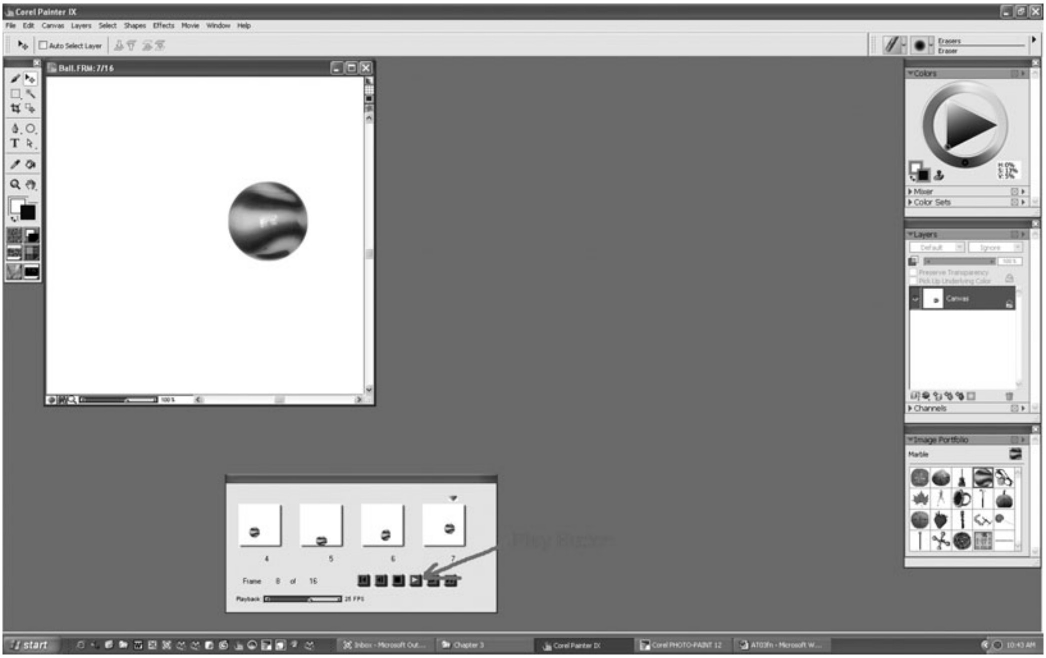


Figure 8.18
Click the Play button to play the animation.

Character Animation

Animating a character is a complex task. Unfortunately, covering every aspect of good character animation is beyond the scope of this book. Instead, it focuses on some basic concepts:

- Comparing motions
- Weight and gravity
- Arcs in animation
- Internal animation

Tip

I highly recommend that you study your favorite animated DVDs to see how animators create the illusion of movement. Try going through the DVD frame by frame; you will be able to see the differences in each frame.

Comparing Motions

Motions vary a great deal. That is, a skip motion is very different from a run motion, and a sneak motion is very different from a casual walk. Even walk motions vary depending on the character's speed. For example, compare the two photos in Figure 8.19. When the model is carrying a heavy duffel bag, he leans forward; this is to balance the added weight of the bag. The more the bag weighs, the more the model will have to lean forward to balance the weight.

Tip

> If you are serious about animation, you should take a look at one of my other books, *The Animator's Reference Book*, also by Thomson Course Technology PTR. In this book, models were photographed from four different angles while performing common game actions. You can use the book to compare motions. The book is also good for studying how balance is maintained and the ways in which body movements differ.

Weight and Gravity

A common problem with novice animators is that their characters lack weight. If you refer to Figure 8.19, you can see how carrying the duffel bag changes the way

Figure 8.19
The model has to lean forward to balance the weight of the duffel bag.

the character walks. But closer study reveals that carrying the duffel bag affects more than just the model's balance. It also affects the effort the model has to make to move. Weight puts more stress on the muscles and joints. This extra stress is evident in the way the model moves in order to carry the heavy bag.

Even without carrying an object, however, a model still has weight, most evident in the model's feet when he or she walks barefoot. Specifically, the foot on the ground will be more compressed than the one in the air. In addition, although casually walking doesn't put much stress on the character's arms, it *does* put stress on the legs. For this reason, the muscles in the legs will tend to show greater degrees of flexing than the arms when a character is strolling along. All this is to say you should observe how weight and gravity affect the human body in motion. Even though some of these effects are subtle, mastering them can bring a huge improvement to animation.

Arcs in Animation

The human body uses a system of bones, joints, and muscles to create movement. Because the joints that anchor the bones act as pivot points, the system always moves in an arc. Tracing these arcs has long been a component of animation; understanding how to do so, however, is no small feat. Figure 8.20 shows the arc of the head and hand in a typical walk sequence. Notice how the motion of movement between frames in the sequence swings in an arc.

Tip

One way to achieve a believable exaggeration in 2D animation is to exaggerate the arcs of human motion. This type of exaggeration will seem natural to the viewer because it fits with the natural movement of the character.

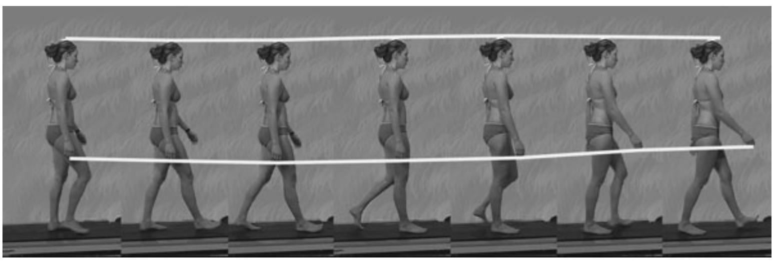


Figure 8.20
All human movement tends to occur in arcs.

Internal Animation

The game artist generally does not deal with movement of a character or object across the screen. Instead, the game programmer usually does that work (although the game artist may need to show the programmer how far to move a character from frame to frame). What the artist is more likely to deal with is the character's or object's *internal animation*—the movement within a character or object itself rather than its movement across a screen. For example, if a character is walking, the movement of the legs is an internal animation, whereas the progress the character makes across the screen is external animation. When an artist animates a character walking in a game, he typically animates the character walking in place. If the artist instead moves the character across the screen, it will make it a lot more difficult for the programmer to move the character in the game.

Creating a Simple Character Animation

Many animations in computer games are *cycled animations*—that is, looping animations that end where they started. For example, most walking and running animation in a game is one full step repeated several times. In the industry, this is referred to as a *walk cycle* or a *run cycle*. This example demonstrates the creation of a walk cycle for a character.

1. In Corel Painter, open the File menu and choose New.
2. In the New dialog box, choose the Movie option button in the Picture Type area, change the frame count to 16, and change the image size to 256×256 (see Figure 8.21). When you're finished, click OK.
3. In the Enter Movie Name dialog box, name the animation Walk, as shown in Figure 8.22.
4. In the New Frame Stack dialog box, change the Layers of Onion Skin setting to 4 and the Storage Type setting to 24 bit color with 8 bit alpha, as shown in Figure 8.23. When you're finished, click OK to continue.
5. This animation uses a character I created earlier. As shown in Figure 8.24, Randy is already in the process of taking a step.
6. Click the Box Selection tool (found on the left side of the screen) and drag a box around Randy to select him (see Figure 8.25).

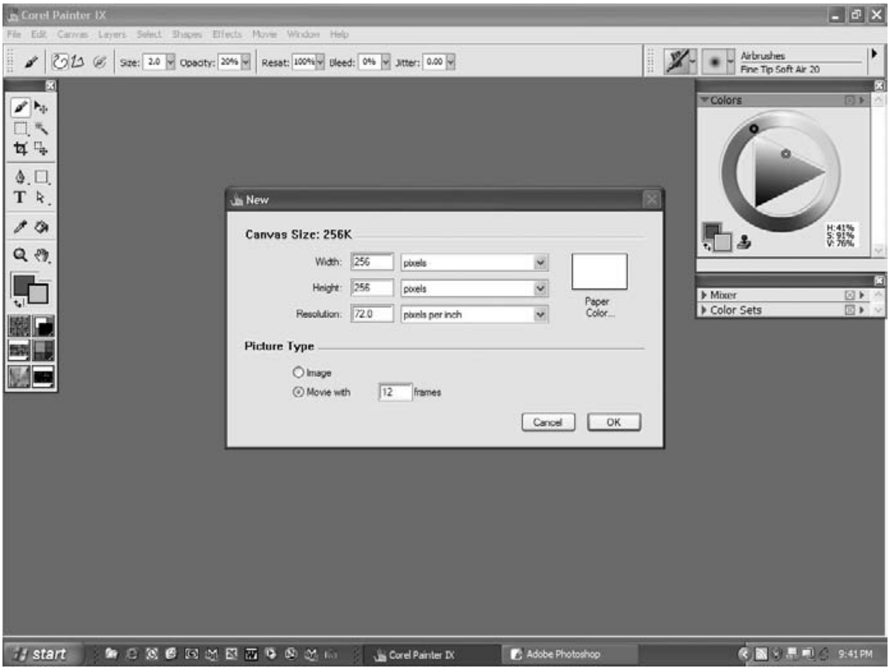


Figure 8.21
Bring up the new animation window.

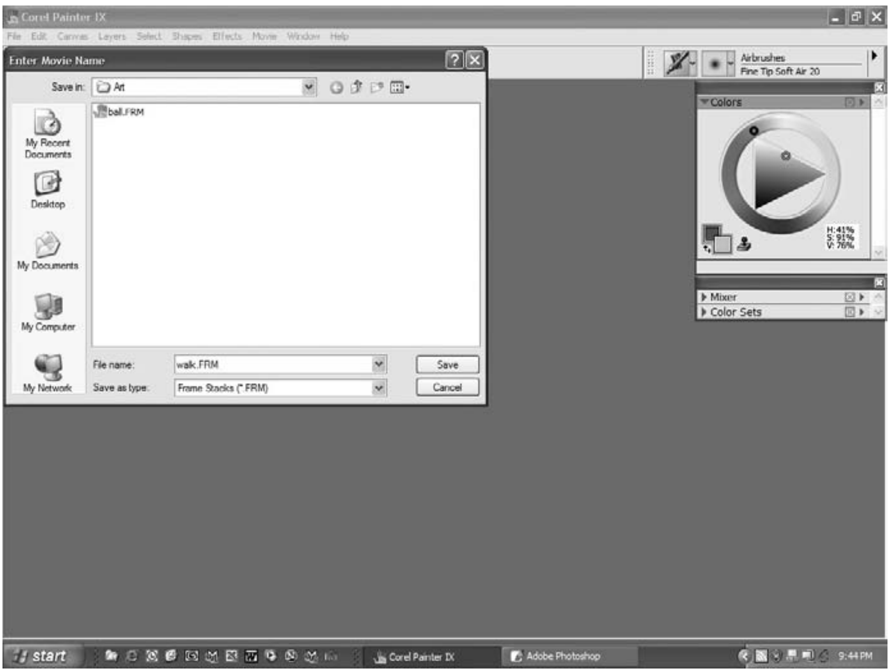


Figure 8.22
The Enter Movie Name dialog box.

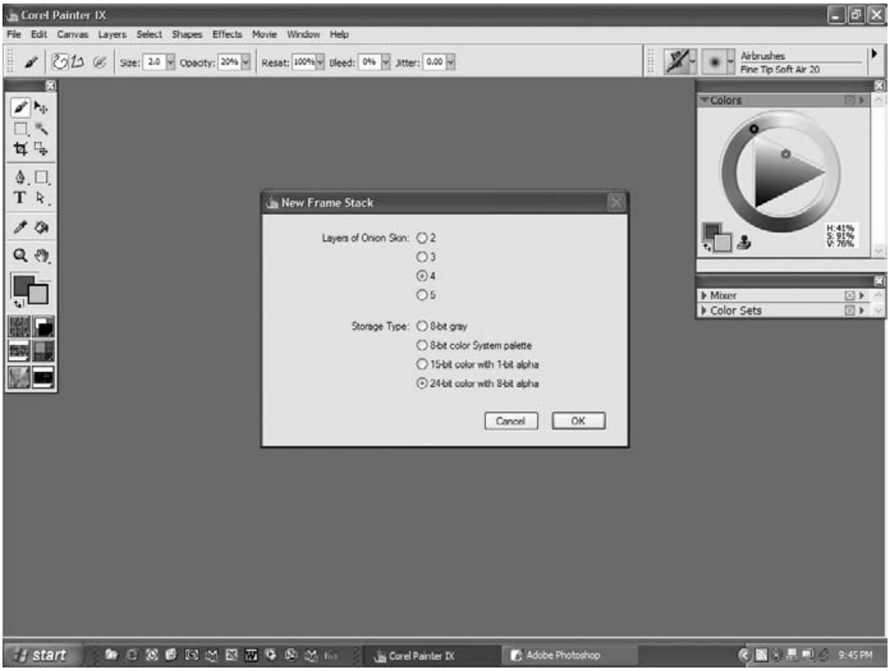


Figure 8.23
The New Frame Stack dialog box.

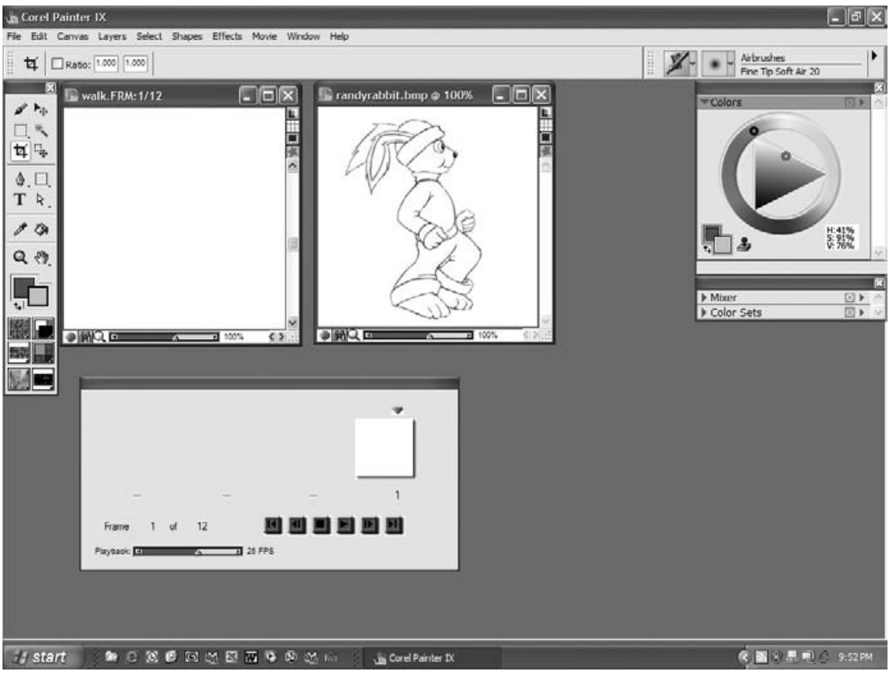


Figure 8.24
A base drawing of the character is loaded as a guide.

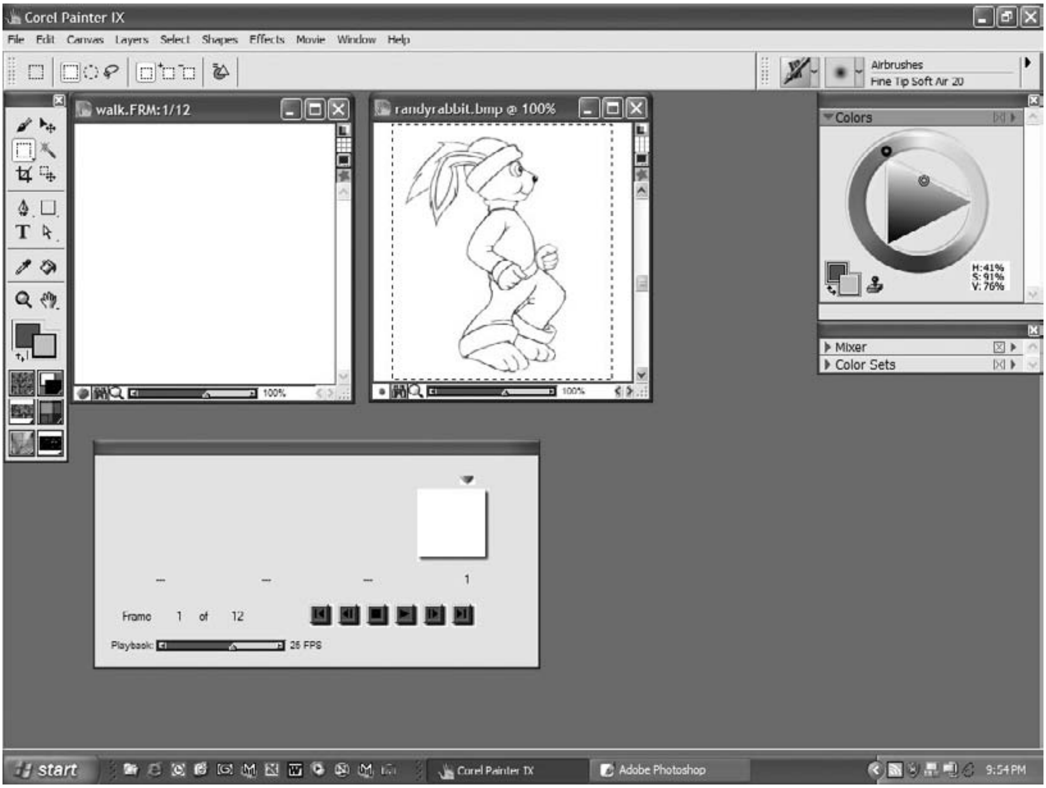


Figure 8.25
Select the character.

7. When the character is selected, simply click him and drag him over to the Animation window. He is now a floating object and can be positioned anywhere on the page; place him in the middle, as shown in Figure 8.26.
8. It is easier to draw an animation if it is larger on screen. To enlarge Randy, slide the blue bar at the bottom of the Animation window to the right until it is set to 200%. (This is the magnification control.)
9. Now grab the lower-right corner of the Animation window and enlarge it to hold the animation.
10. Using the animation controls, move to frame 2. The drawing of Randy is now flattened onto the frame 1.
11. Because you want to change Randy's position in frame 2, open the Layers menu and choose Delete to delete the floating drawing (see Figure 8.27).

Figure 8.26
Move the selected character to the Animation window.

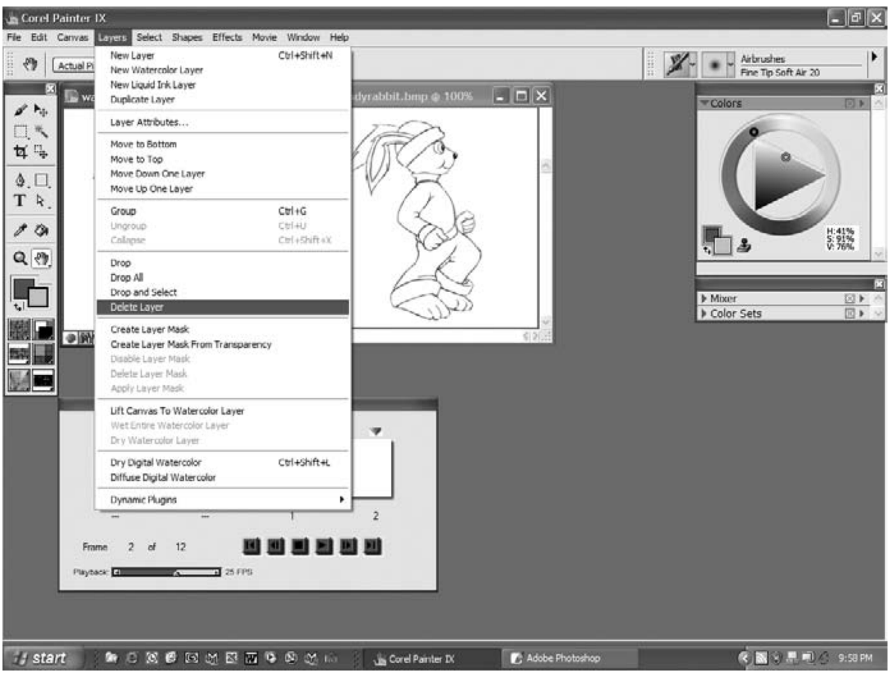


Figure 8.27
Delete the floating layer from frame 2.

Figure 8.28
Turn on the Tracing Paper feature.

12. Click the Toggle Tracing Paper button in the upper-right corner of the Animation window to turn on the Tracing Paper feature (see Figure 8.28).

13. Now that you can see the first drawing through the second frame, you can use it as a guide for drawing the second frame of the animation. Draw the second frame with Randy's feet and hands moved as shown, raising him slightly to give him a bounce as he walks (see Figure 8.29).

14. Continue to draw new frames of the animation of Randy's walk using the drawings in the frames below as a guide (see Figure 8.30).

Figure 8.29
Frame 2 of the animation is drawn.

Figure 8.30
Draw frames 1–4 of the animation sequence.

Timing

One of the biggest challenges of animation is timing. In a walk cycle, the rate of movement is fairly constant, meaning that the act of walking generally does not have uneven movement. In most cases, the animator can divide the distance of movement between each frame evenly. In a 12-frame animation, frame 1 and 7 will look much the same with the only exception being that the left and right sides of the character will be reversed (see Figure 8.31).

Figure 8.31
Frame 1 and frame 7.

The same thing will be true for frames 2 and 8; frames 3 and 9; frames 4 and 10; frames 5 and 11; and frames 6 and 12. If the artist does a good job with the first half of the walk cycle, the second half becomes easier because the first half can be used as a template. Look at Figure 8.32 and notice the similarities between frames.

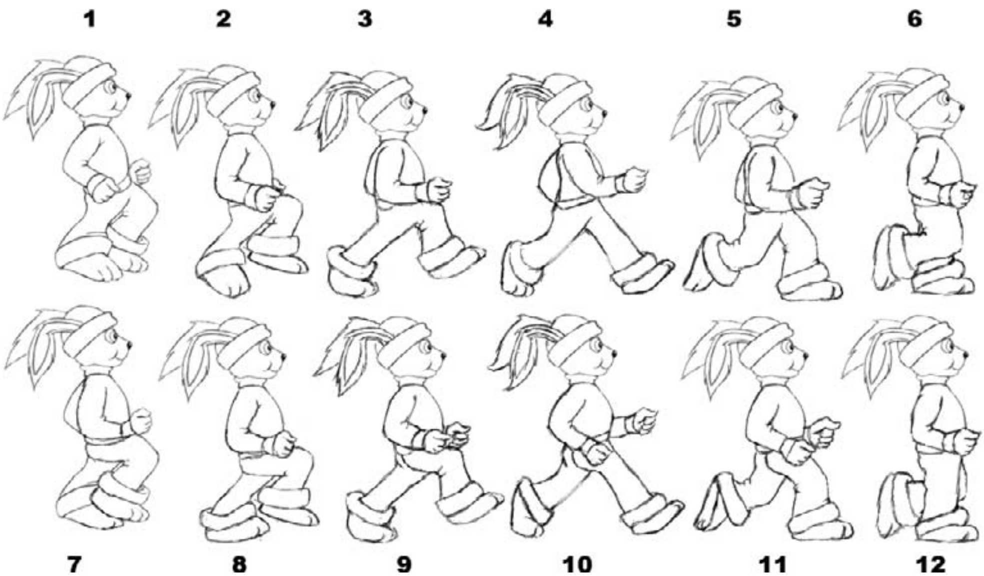


Figure 8.32
Notice the similarities between frames of animation.

Figure 8.33
Frame 12 of the animation sequence.

15. Continue drawing each frame. When you reach the last frame, compare each frame as in Figure 8.33. Review the motion by incrementing through the animation one at a time.

Caution

Be careful to only go to 12. If you increment beyond 12, Corel Painter will automatically add unwanted frames.

16. Click the Play button. The animation will run quickly, but you will be able to see your work (see Figure 8.34). You can slow the animation by adjusting the frame slider on the animation menu.

3D Animation

With your model rigged, you are now ready to learn how to make him move. Maya has a great history as an animation tool, and one of its greatest attributes is how easy it is to animate complex characters in 3D. Although there are a few

Figure 8.34
Play the animation.

similarities, 3D animation is different in many ways from 2D animation. In 2D animation, each frame of animation is a different picture. In contrast, 3D animation characters and objects are moved in the 3D world and their motion is stored in a *motion file* that becomes part of the Maya file. Motion files can contain data on almost every attribute of a 3D model, including translation, rotation, size, color, and many others. The animation possibilities are almost endless.

In this section, you will learn how to use some of Maya's powerful 3D animation features to apply motion to your character. Specifically, you will learn the following:

- Maya's animation controls
- Moving objects
- Path Animation
- Creating a simple character animation

Maya's Animation Controls

Animation in Maya is controlled primarily with use of the animation controls at the bottom of the screen. Figure 8.35 highlights the animation controls.

- **Current Frame indicator.** The Current Frame indicator shows the animation frame that is currently shown in the view screen.
- **Timeline.** This shows in a visual sequence the available frames in an animation. The Current Frame indicator slides along the Timeline.
- **Current Frame Number field.** Type a new number in this box to change the current frame.
- **Playback controls.** These are used to view animations or to navigate through an animation. These controls are similar to the controls on a DVD player.
- **Start time.** This indicates the beginning frame of an animation. Note that the start time does not need to be 1; it can be any number.

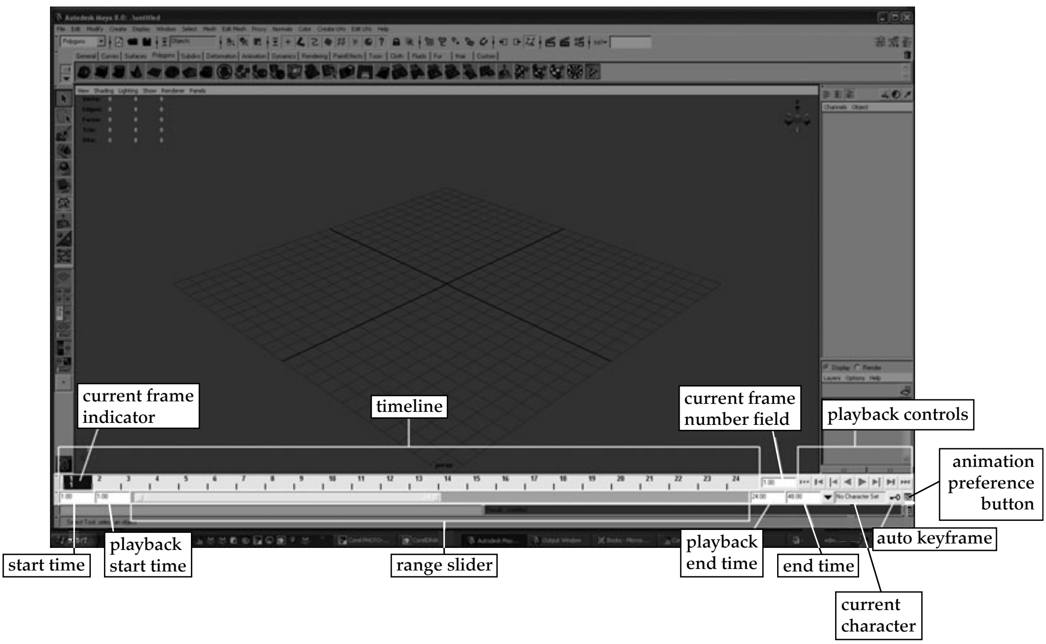


Figure 8.35
Maya's animation controls are located near the bottom of the interface.

- **Playback start time.** This indicates the start time of a playback sequence.
- **Range slider.** This shows the playback range of an animation within the total number of frames.
- **Playback end time.** This indicates the last frame of the playback range.
- **End time.** This indicates the last frame of the animation.
- **Current Character indicator.** This shows what the current character is.
- **Auto Keyframe.** This indicates whether the Automatic Keyframe feature is turned on or off.
- **Animation Preference button.** Click this button to open a window that lets you change the frame rate, timelines, and other playback options.

Take some time to familiarize yourself with the controls. Have this list handy as you start to work on your animations so you can find the controls mentioned in the exercises in this book.

Moving Objects

Object animation is the simplest form of 3D animation. All it requires is to set keyframes in the locations where you want the object to move. When the animation is played back, the object will then travel to each location.

Note

Unlike 2D animation, where you have to draw every in-between frame, Maya creates them for you and even allows you to manipulate their timing. Let's try it out with a simple ball animation.

1. Create a polygon sphere in the y axis that has a radius of 3, as shown in Figure 8.36. Make the subdivisions around the axis 16 and the subdivisions along the height 8, as shown.
2. Build a single polygon for the ground that is 32 units square, as shown in Figure 8.37.
3. Move the ball up 3 units so it is sitting on the ground.

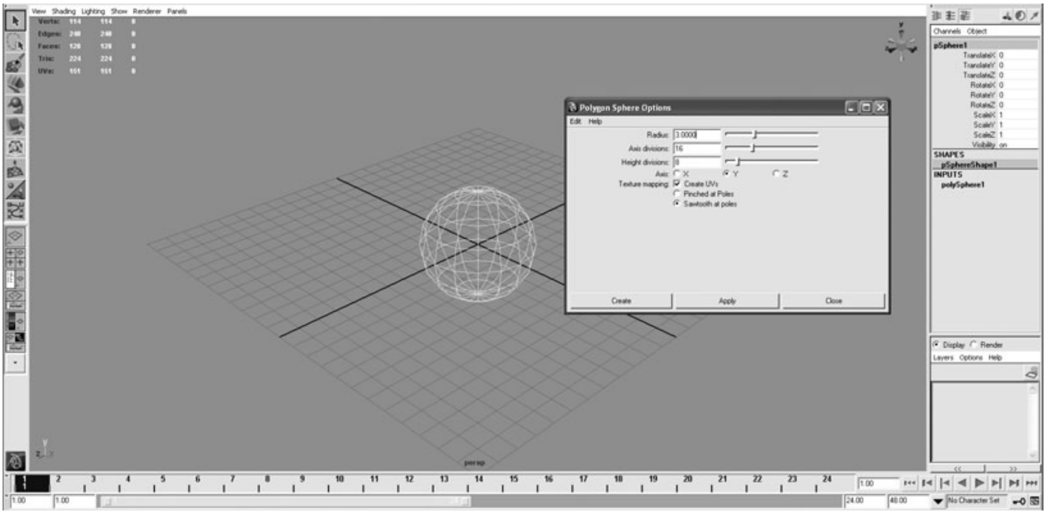


Figure 8.36
Create a polygon sphere.

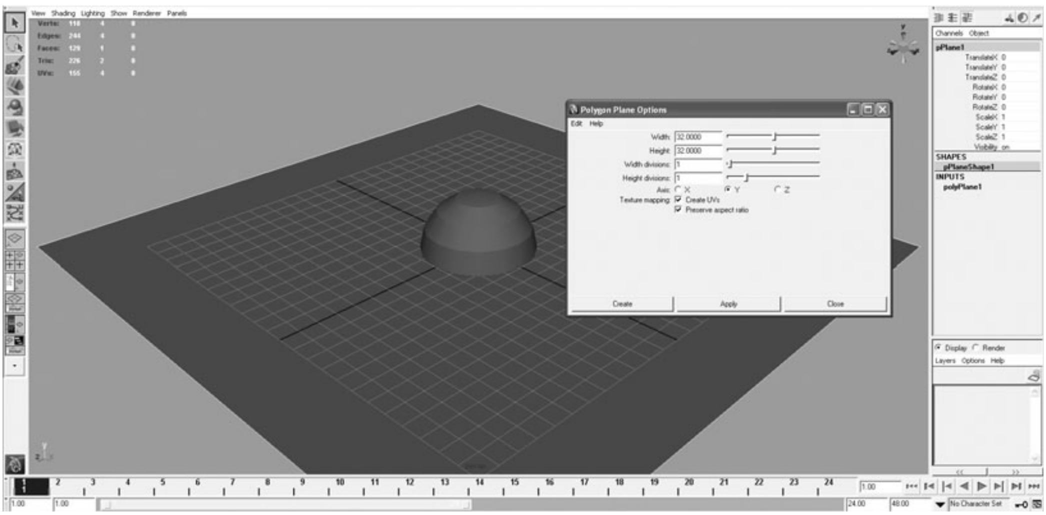


Figure 8.37
Create a polygon for the ground.

4. Create a new material using the default checkerboard texture shown in Figure 8.38.
5. Apply the new material to the ball.
6. Create another new material, but this time, use the Fractal texture and apply it to the polygon plane, as shown in Figure 8.39.

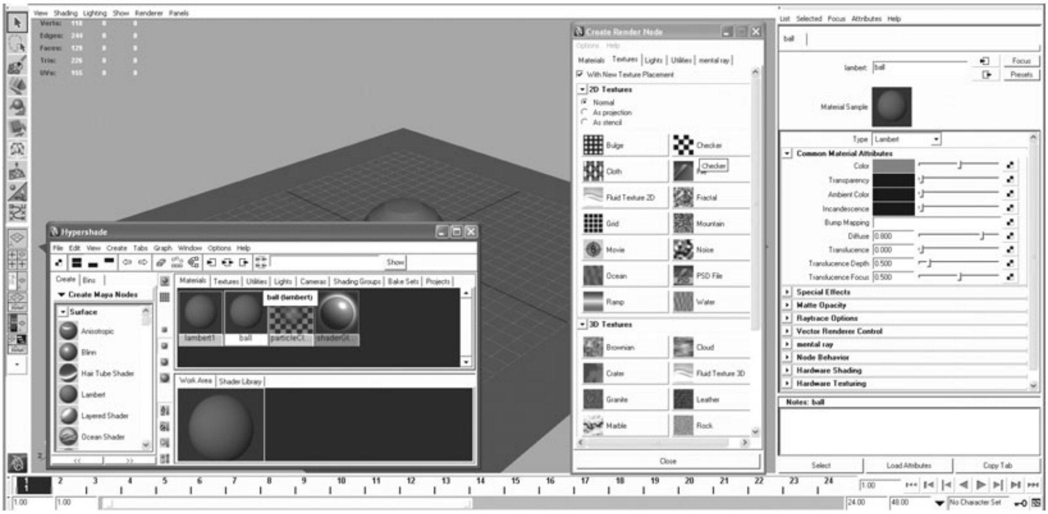


Figure 8.38
Choose checker from the Create Render Node dialog box.

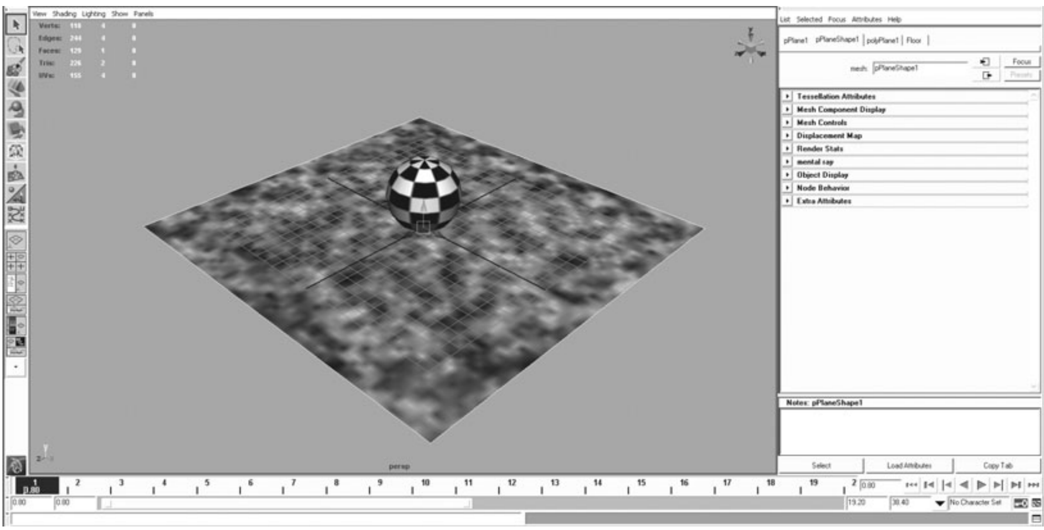


Figure 8.39
Give the ground a fractal texture.

7. Use the Polygon Planar Projection tool to scale the fractal texture as shown in Figure 8.40.

8. Go to the Front view and move the ball up in the y axis to 24 and over in the x axis −20 as shown in Figure 8.41.

9. If you haven't already done so, change the Main menu set to Animation.

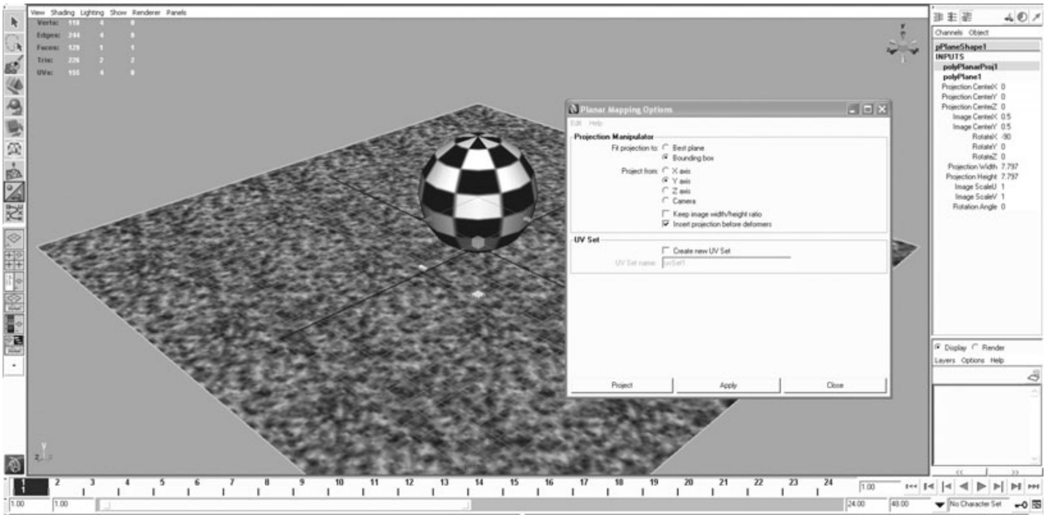


Figure 8.40
Scale down the fractal texture on the polygon plane.

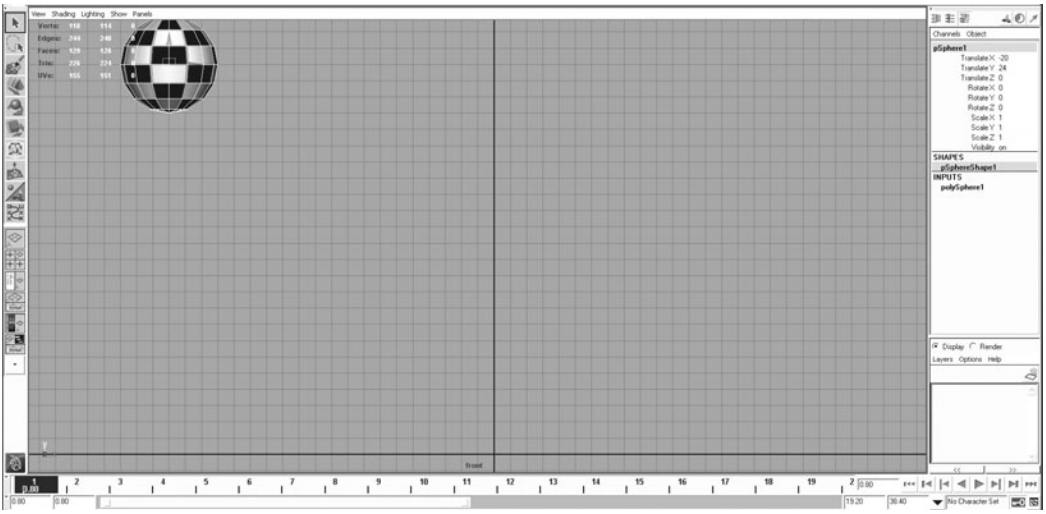


Figure 8.41
Move the ball to the upper-left part of the screen.

10. There are several ways to set keyframes in Maya. If you go to the Animate menu, you will notice that the first item in the menu is Set Key (see Figure 8.42). You should also notice that there is also a hot key associated with the function. To set a keyframe, all you need to do is press the S key. Set a keyframe for frame 0.

Figure 8.42
Set a keyframe.

Note

Setting a keyframe is kind of like taking a picture of the current scene. Maya remembers the positions of all the selected elements in the scene. Items that are not selected when the keyframe is set will not have a key associated with them. You can set keys only for selected items. This will become very important later when you start working on more complex character animation.

11. Another easy way to set keyframes is to use the Auto Keyframe function. You can toggle this function off and on by clicking the Auto Keyframe toggle in the lower-right corner of the screen. When it is on, the toggle turns red. Turn it on now.

12. Move the Current Frame indicator to frame 9.

13. Move the ball to −1 in the x axis and 3 in the y axis.

14. Move Current Frame indicator to frame 10. Notice there is a red marker on frame 9. These red markers indicate that there is a keyframe on that frame.

15. Move the ball to 0 in the x axis and 2.5 in the y axis. Use the scale tool to squash the ball so it looks like it is hitting the ground as shown in Figure 8.43. (You will need to enlarge the ball a little so it looks like it is the same size or it will appear to shrink.)

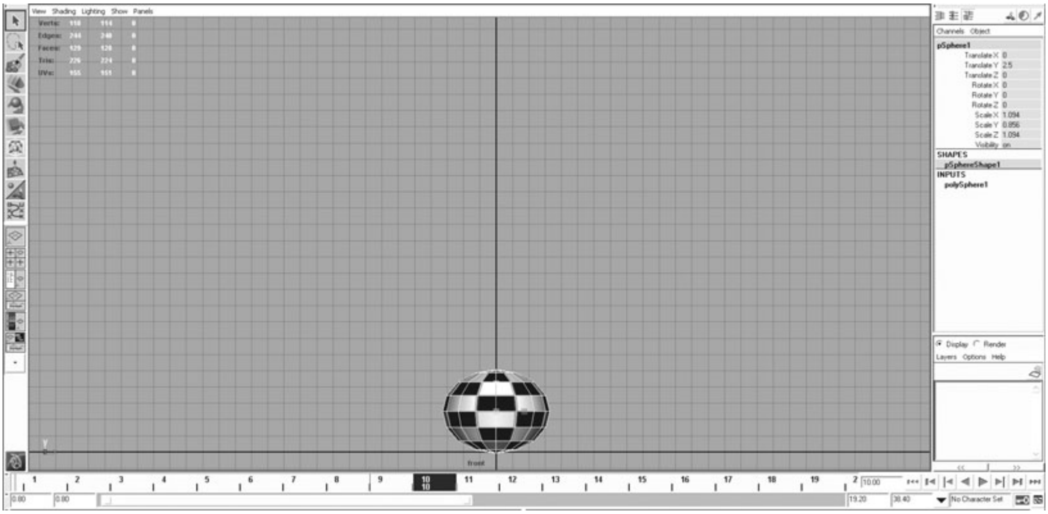


Figure 8.43
Make the ball look like it is hitting the ground.

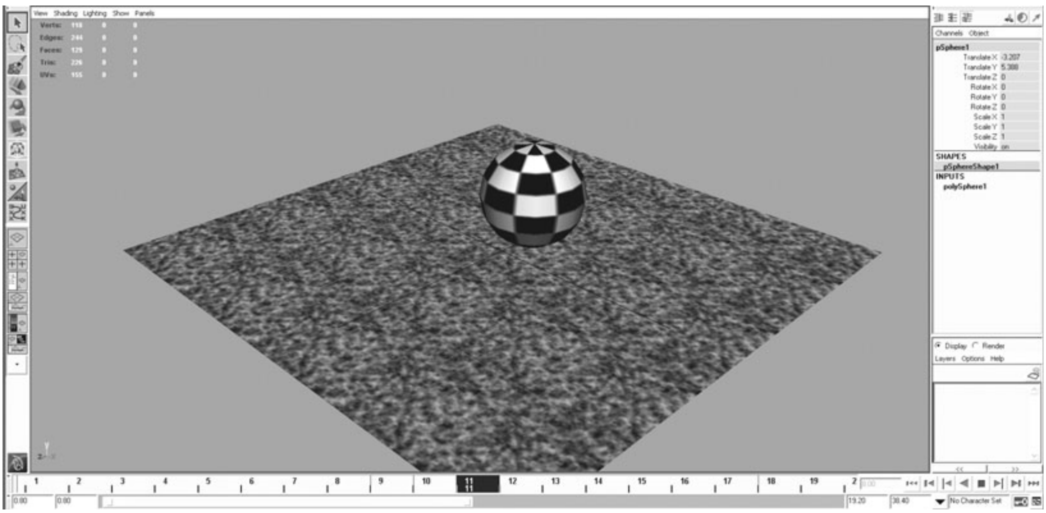


Figure 8.44
View your animation.

16. Move the Current Frame indicator to frame 18.
17. Move the ball to 20 in the x axis and 20 in the y axis.
18. Go to the Perspective view and click the Play button on the Playback controls (see Figure 8.44). The ball should look like it is bouncing off the ground.

That wasn't too hard, now was it? Now that you know how to move things around and record keyframes, you can experiment with a variety of shapes. Although most of the animation you do for games will not deal with external motion like this animation does, there are instances in which you will use external animation. For example, your game may call for an animation sequence to be played at the beginning of the game or a level. These sequences are essentially like creating a movie and are referred to as *FMVs* in the industry, which is short for *full motion video*. The player has no real control of an FMV; all movement inside it is controlled by the animator.

Path Animation

Another way to move an object in Maya is to have it follow a path. *Path animation*, especially useful when animating objects that need to travel along a specific path (think a car on a road or an airplane coming in for a landing), attaches an object to a curve. In some games, the animator sets up the paths and the timing, and the programmer uses the path for the animation of NPCs. An example of this is in racing games, where there is an optimum path for all the cars to travel on the racetrack. The programmer uses the optimum path and sets a variance to it. The cars then travel within that variance, making them look like they are all traveling on different paths. The better drivers will be closer to the optimum path.

To demonstrate how to set up a path animation in Maya, use the bouncing ball animation from the last section.

1. Create a NURBS circle with a sweep of 360 degrees and a radius of 50, as shown in Figure 8.45. This will be the path.

2. Select the Polygon plane. Then, while pressing the Shift key, click the circle to select it.

3. Open the Animate menu, choose Motion Paths, and select Attach To Motion Path to open the Attach to Motion Path Option dialog box.

4. Click the Time Slider option button next to Time Range to select it.

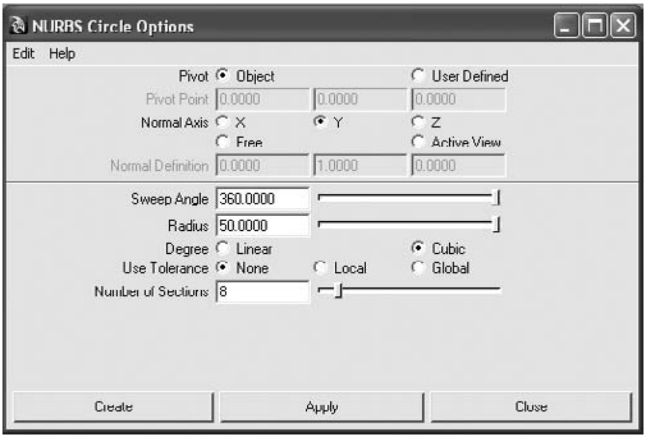


Figure 8.45
Create a NURBS circle for the path.

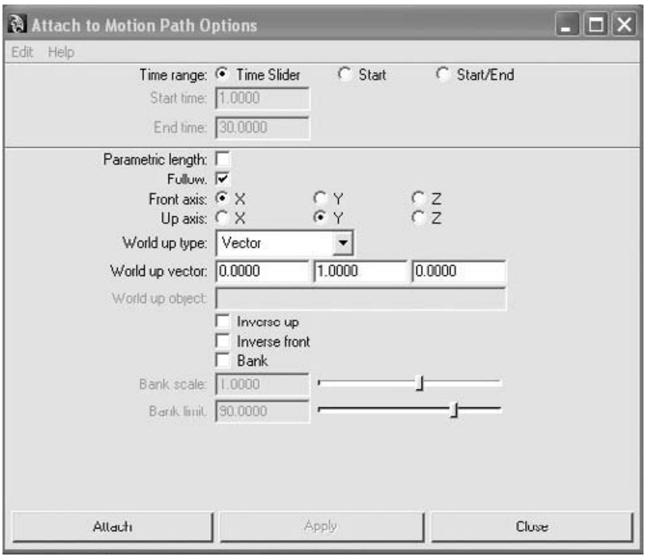


Figure 8.46
Attach the plane to the path.

5. Check the Follow checkbox to have the plane follow the path.
6. Next to Front Axis, click the X option button.
7. Next to Up Axis, click the Y option button.
8. Click Attach to attach the plane to the path (see Figure 8.46).

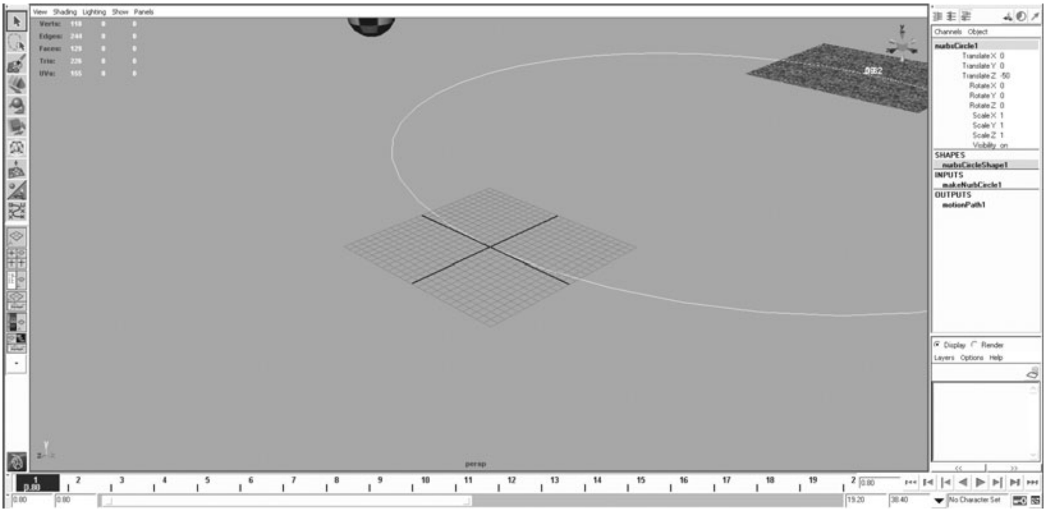


Figure 8.47
Move the circle so the plane will match up with the ball when it hits the ground.

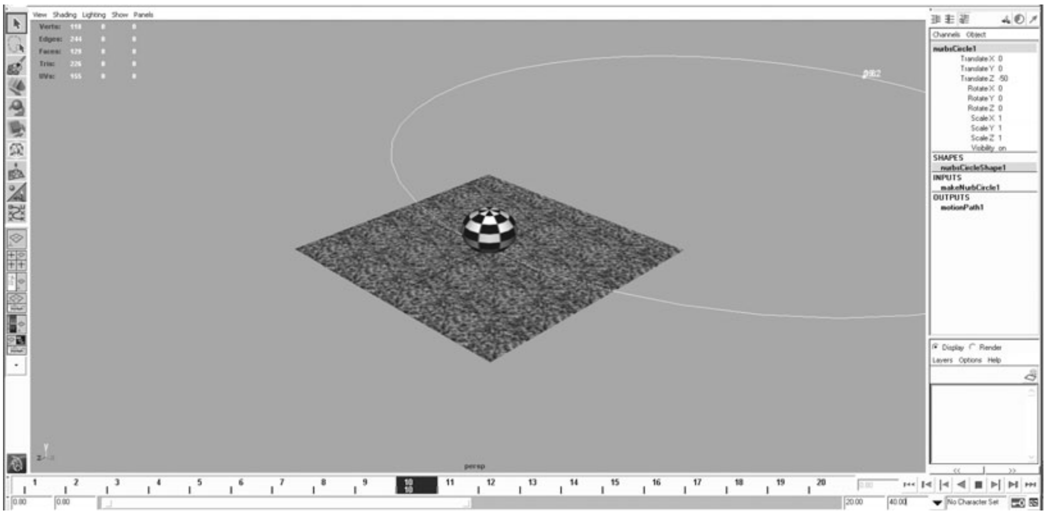


Figure 8.48
Play the animation.

9. Select the circle and move it −50 in the z axis, as shown in Figure 8.47.
10. Click the Play button and watch what happens. The ball should hit the ground just as the ground comes underneath it, as shown in Figure 8.48.

This example is a very simple application of path animation in Maya. Try a few experiments with the features associated with paths until you become comfortable with them.

Simple Character Animation

Now you are ready to begin animating your character. To begin, click the Auto Key button to enable it. This automatically sets keys to the attributes that you animate. Maya treats every aspect of each part of a bone hierarchy as a separate element that can be keyed.

When Auto Key is on, it will set keys for any aspect of a bone that changes and ignore those that don't change. It is very important that you remember this when animating. There are times when you will want a key set even if that part of the bone didn't move. For example if you are having a character turn his head at a particular time in your animation, you may want to have a key set for the rotation of the head just before it is time for him to turn his head. If you didn't set a key there and just used Auto Key, the head would start moving at the last set key rather than when you wanted it to turn. A good rule it to set a key for any element that you want to move at the beginning and end of each movement.

Now let's get started animating.

Posing the Model

Keyframe animation basically involves moving the model from one pose to another, so start by learning how to pose your character. Because a keyframe is a single moment in time, the idea behind keyframe animation is to pose the character in the extremes of each motion and then allow Maya to fill in the frames between the poses. Right now, the character is in the T pose that we used for building and rigging. Although the T pose works great for building a character, it isn't a very natural looking pose.

1. Click the bottom-most Quick Layout button to change the screen to the three-panel animation layout (see Figure 8.49).

2. Change the first frame from 1 to 0. You can now place the T pose in frame 0 and start your animation on frame 1.

3. In the Hypergraph panel, select all the nodes in the bone hierarchy and press the S key to set a key on frame 0, as shown in Figure 8.50.

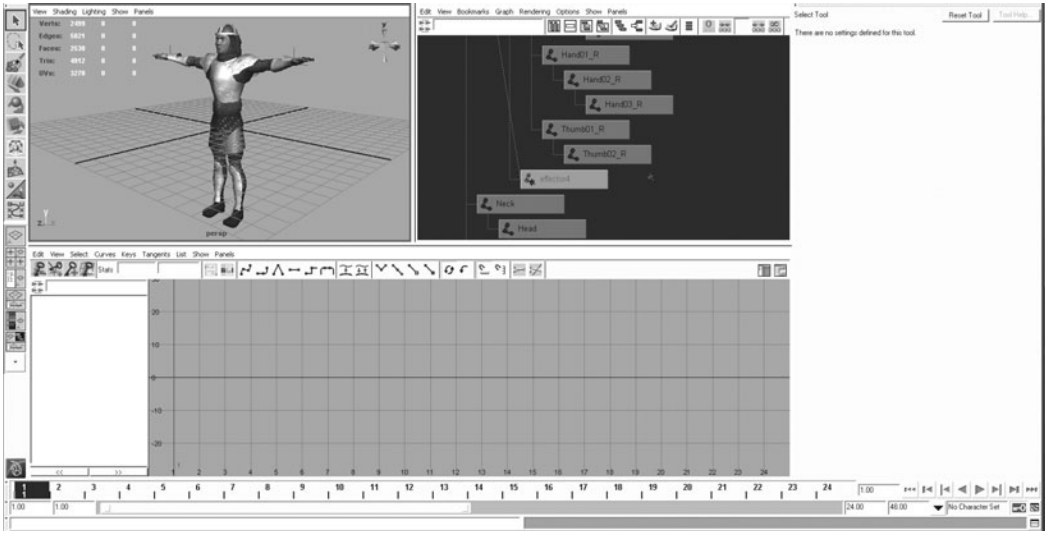


Figure 8.49
Change the screen to the three-panel animation screen layout.

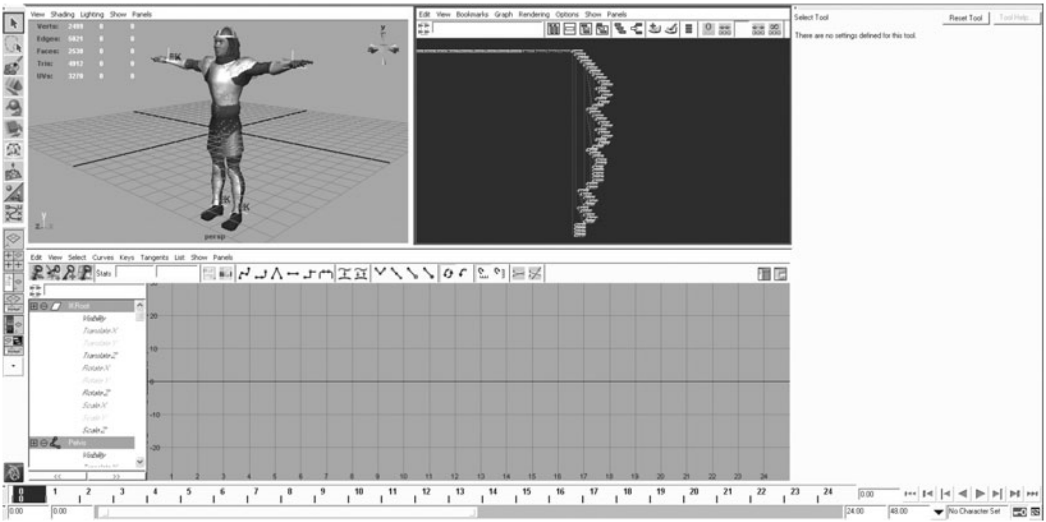


Figure 8.50
Set a key on frame 0.

4. Move the IK handle for the wrists down so that the arms are by the character's side, as shown in Figure 8.51.
5. Use the Rotate tool to rotate the armor joints to about −25 in the z axis, bringing them down over the shoulders as shown in Figure 8.52.

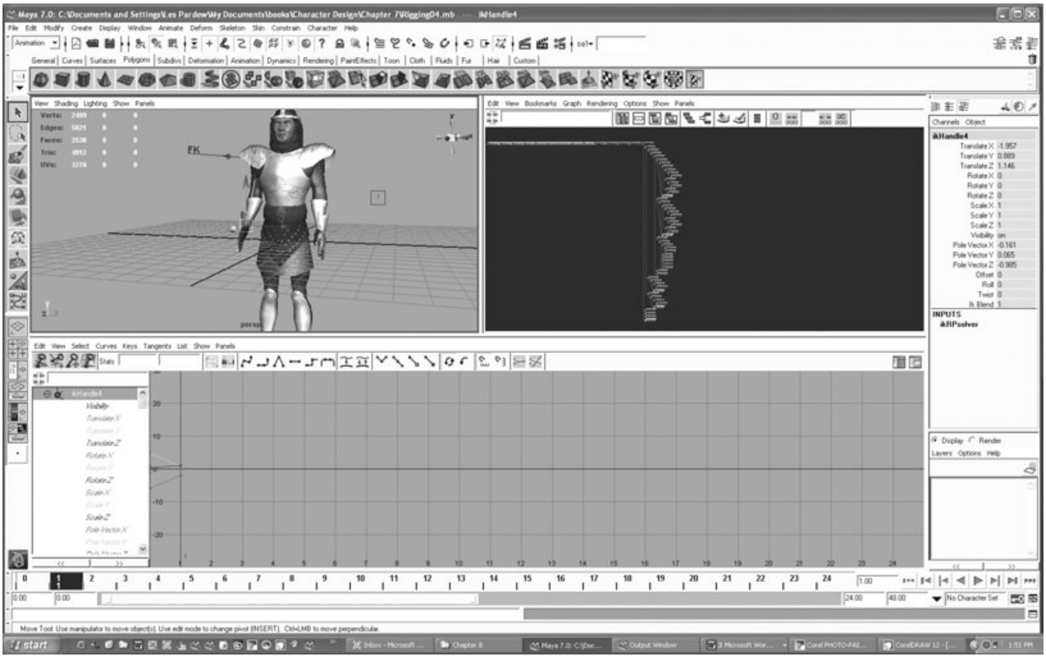


Figure 8.51
Put the character's arms down by his sides.

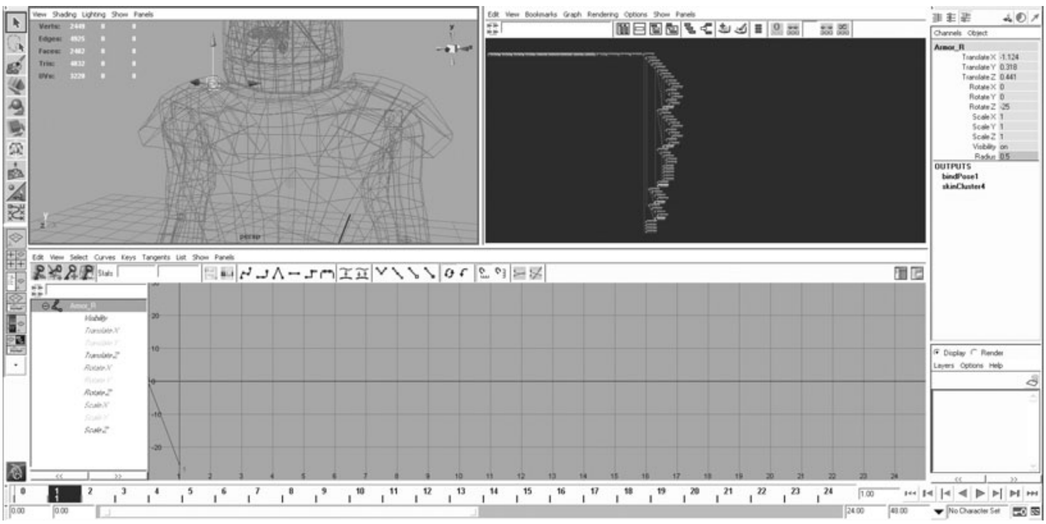


Figure 8.52
Lower the shoulder plates.

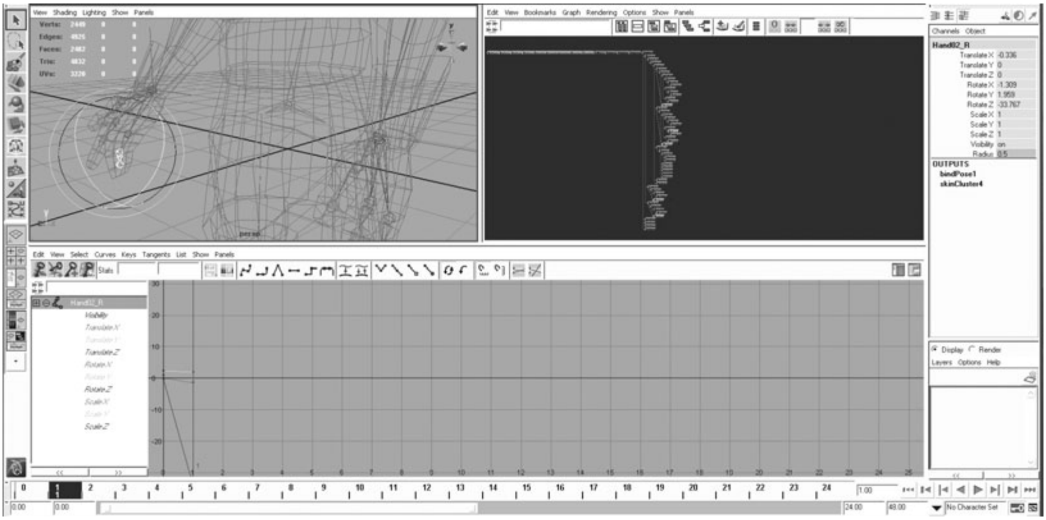


Figure 8.53
Rotate the finger joints to give them a more relaxed position.

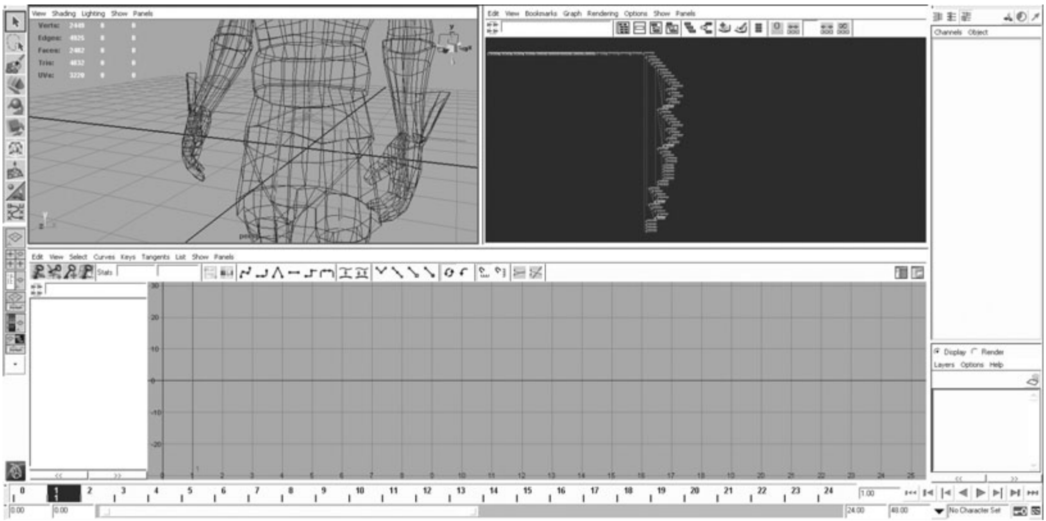


Figure 8.54
Both hands should look relaxed.

6. Because the character's fingers are sticking straight out, they look a little stiff. You can rotate the joints in the hand to bend the fingers as shown in Figure 8.53.

7. Rotate the finger and thumb joints to make them look more relaxed on both hands as shown in Figure 8.54.

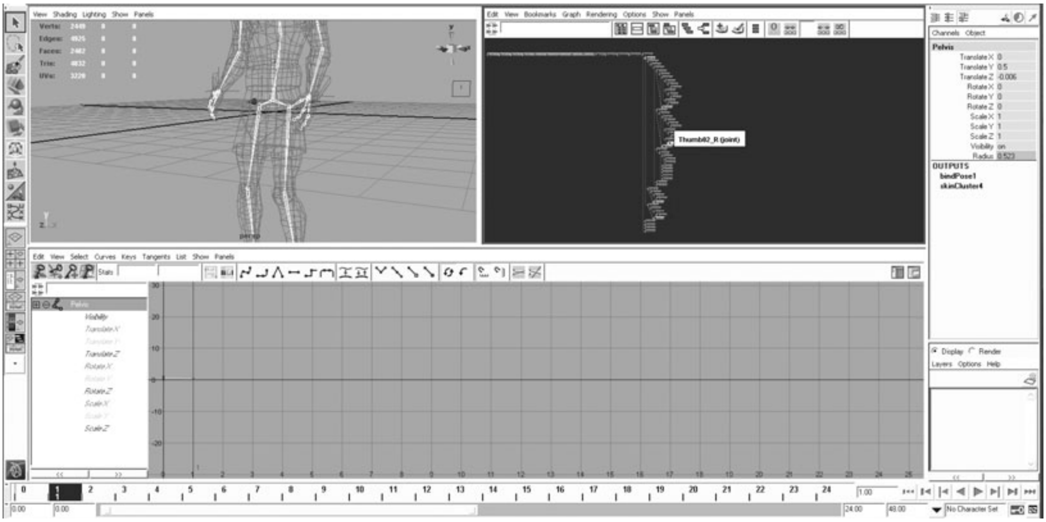


Figure 8.55
Lower the pelvis.

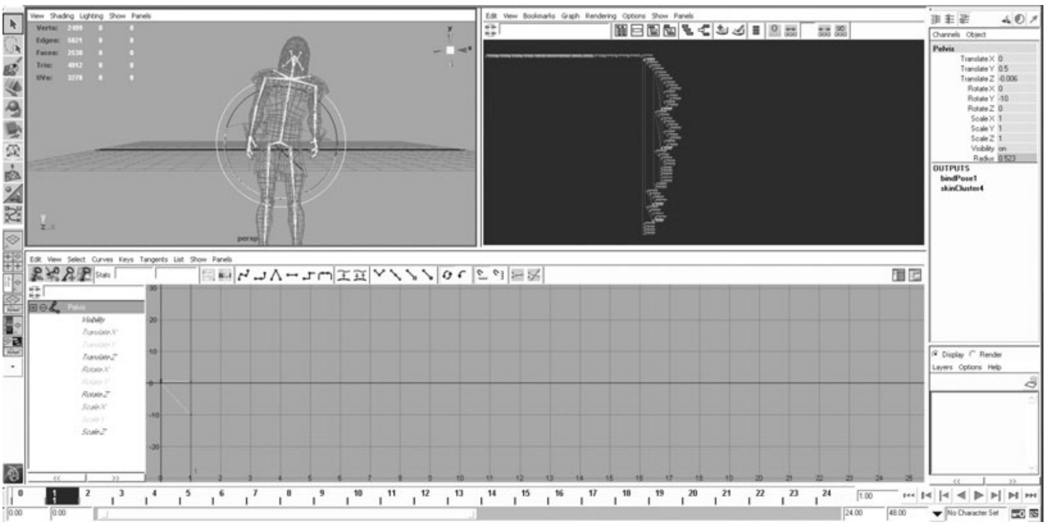


Figure 8.56
Tilt the hips to one side.

8. The beginning pose will have the character stand with most of his weight on one leg. Select the pelvis joint and lower the character a little so that both knees are bent, as shown in Figure 8.55.
9. Rotate the pelvis −10 degrees in the y axis to give the hips a little tilt, as shown in Figure 8.56.

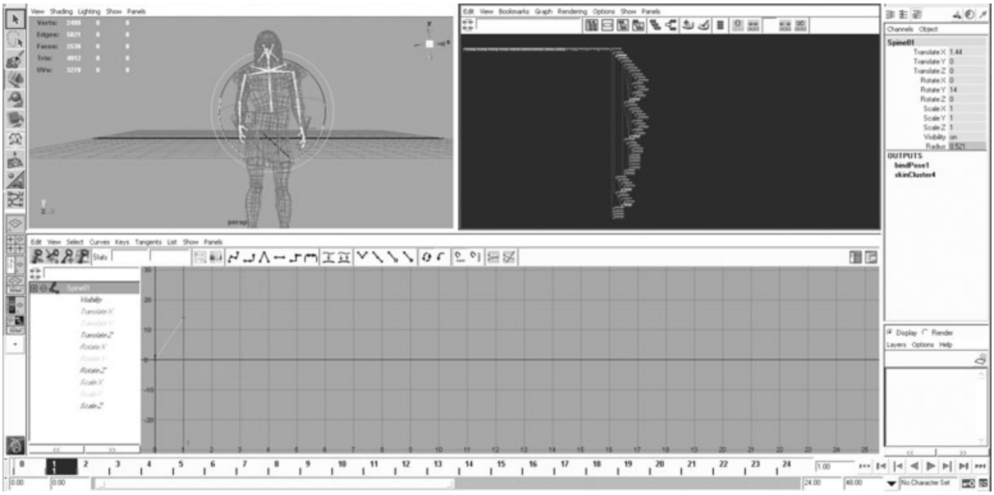


Figure 8.57
Tilt the upper torso in the opposite direction from the hips.

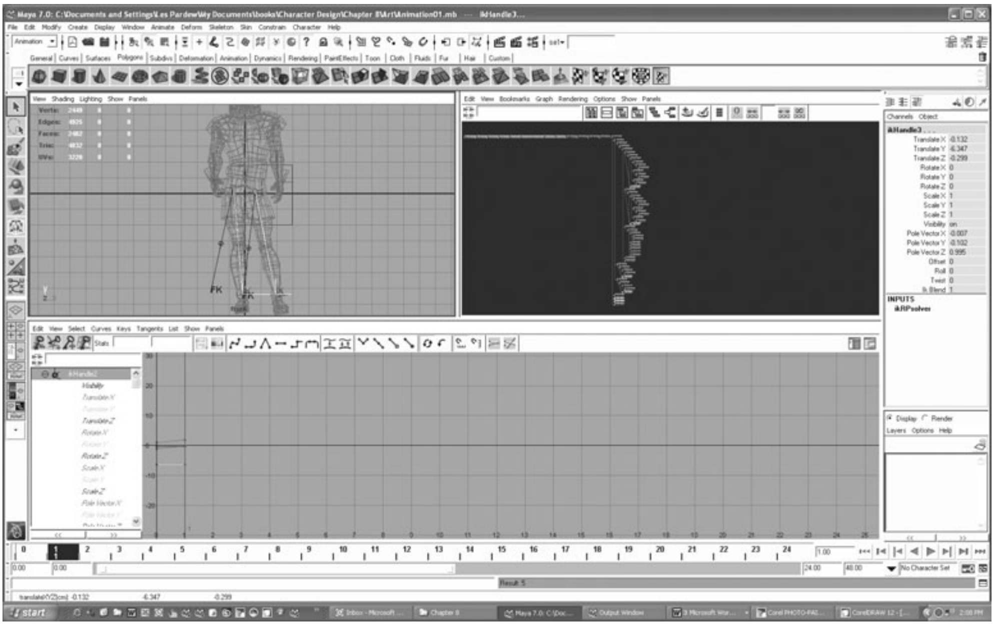


Figure 8.58
The character's weight should be centered over his right foot.

10. Counter the tilt of the hips by rotating the spine1 joint 14 degrees in the y axis, as shown in Figure 8.57.

11. Select both IK handles for the ankles and move them so that the characters weight is primarily over his right leg, as shown in Figure 8.58.

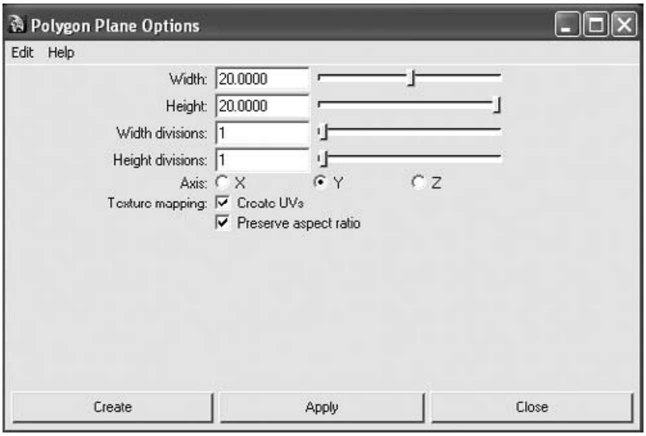


Figure 8.59
Create a polygon plane for the ground.

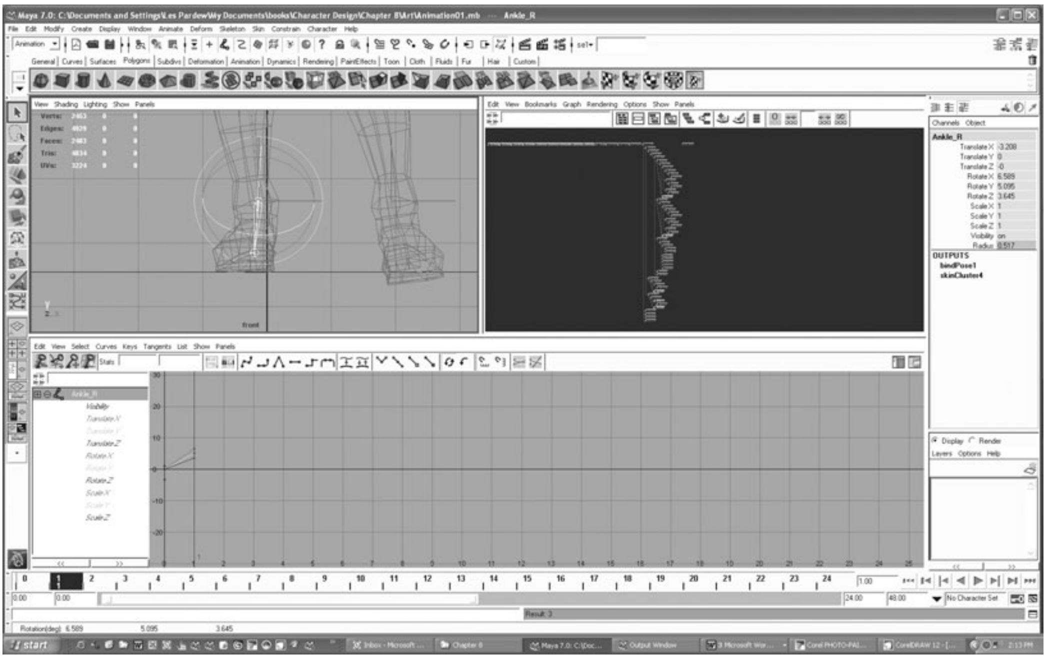


Figure 8.60
Plant the right foot on the ground.

12. Create a single 20 × 20 polygon plane for a ground plane, as shown in Figure 8.59. Move the ground plane down to the character's feet.

13. Position the right foot so it looks planted on the ground (see Figure 8.60).

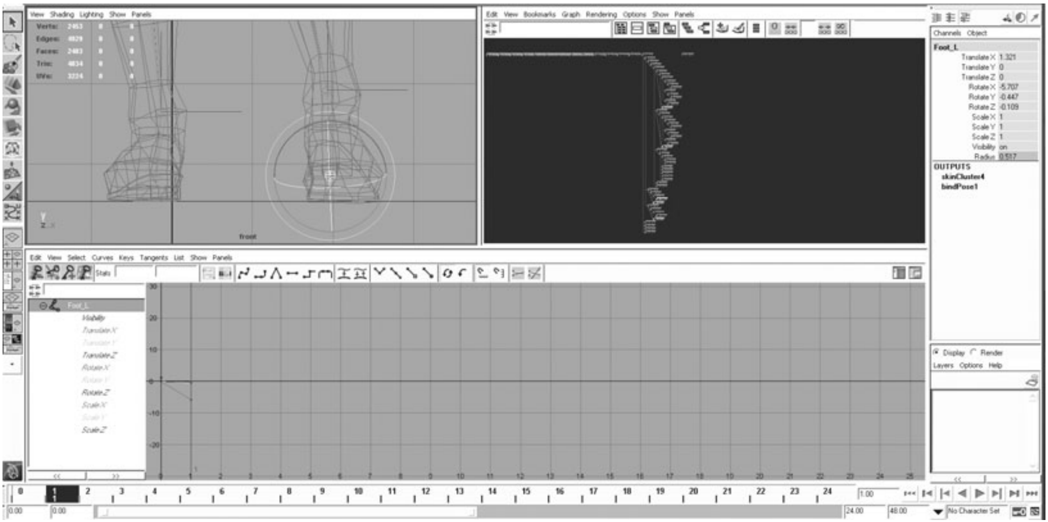


Figure 8.61
Raise slightly the heel of the left foot.

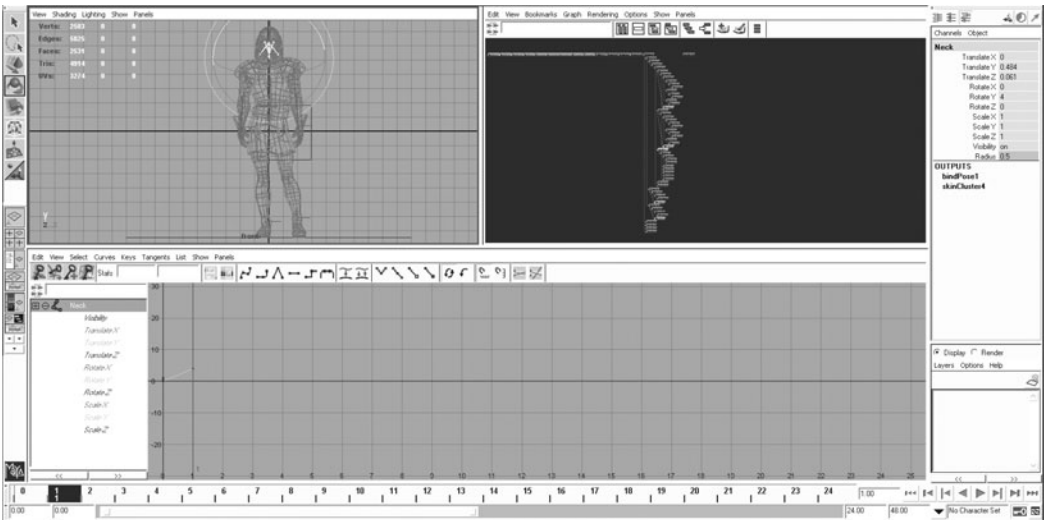


Figure 8.62
Rotate the neck and head.

14. The left foot can be raised slightly at the heel as shown in Figure 8.61.
15. Rotate the neck joint −4 in the y axis to straighten it as shown in Figure 8.62.

The character is now in a relaxed standing pose with his weight primarily on his right leg. This will be the beginning pose for your animation. Because it is the

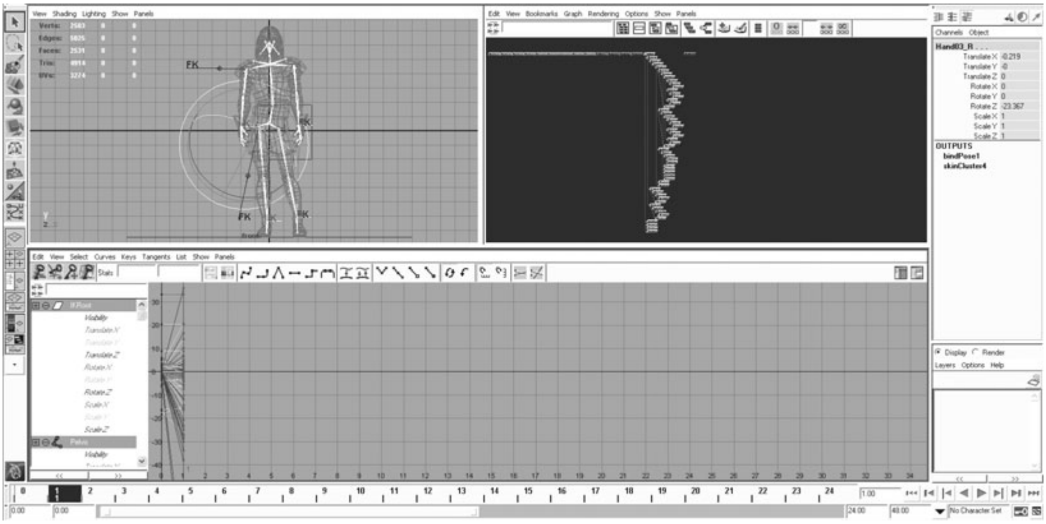


Figure 8.63
Set a key at frame 1.

beginning pose, you need to set a key at this frame for all the nodes in the bone tree. Select all the bone in the node tree and press the S key to set a keyframe at frame 1 (see Figure 8.63).

If you noticed the Graph editor, you will see that there are lines between frame 1 and frame 0. These lines represent motion over time. If the line is flat, there is no motion for that node. If the line moves up or down, there is motion for that node. The more the line goes up or down, the more distance there is in the motion of the joint from one keyframe to another.

Animating a Salute

The relaxed standing pose will be the base pose for the character. From this pose, he will perform many of his movements. In this section, you will create a simple salute animation for your character, starting with him in the relaxed pose position. The next pose that you will use for a keyframe will have the character stand at attention as if an officer just walked by. The attention pose is more like the T pose than the relaxed pose.

Maya allows you to copy and paste keyframes from one key to another. To do so, go to the frame you want to copy (in this case, frame 0), and right-clicking the

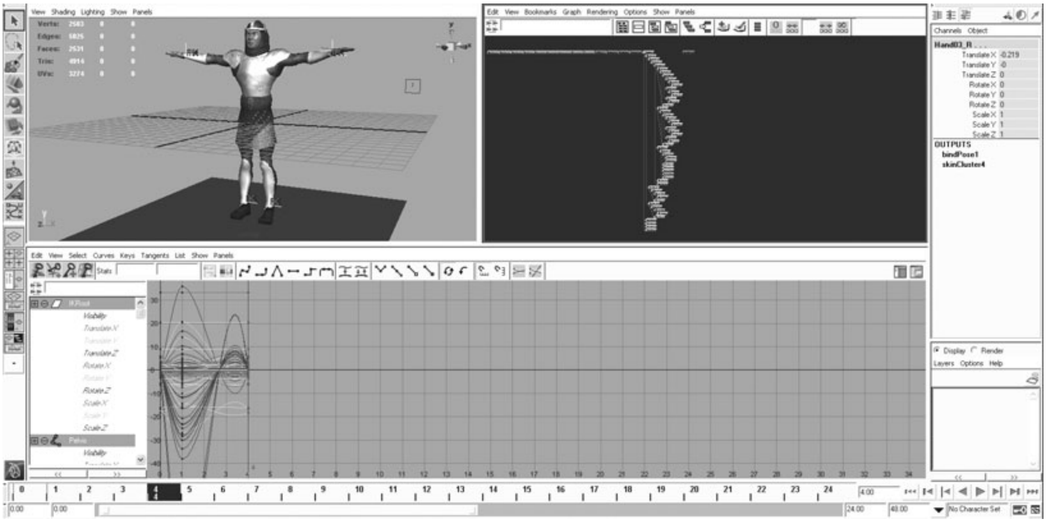


Figure 8.64
Paste the T pose into frame 4.

Current Frame indicator to open the context menu; then choose the Copy command. Before you copy the frame, however, you should make sure you have selected all the bone nodes in the bone tree in the Hypergraph.

1. Start by moving the Current Frame indicator to frame 4. This will be your next keyframe.

2. Right click the Current Frame indicator and select Paste from the menu to paste the key from frame 0 to frame 4, as shown in Figure 8.64.

3. Lower the character arms so they're straight down by his sides in an attention pose, with the hands and fingers straight and rigid as shown in Figure 8.65.

4. Move to frame 10. This is where the character will be holding a fist to his heart in a salute, as if an officer just walked up to him. Notice as you move the character's arm up to his heart that the elbow sinks into the chest. To rectify this, click the Show Manipulator tool to launch the Pole Vector tool, which can change the rotation of the arm within the IK handle. Click the blue circle and rotate the elbow out as shown in Figure 8.66.

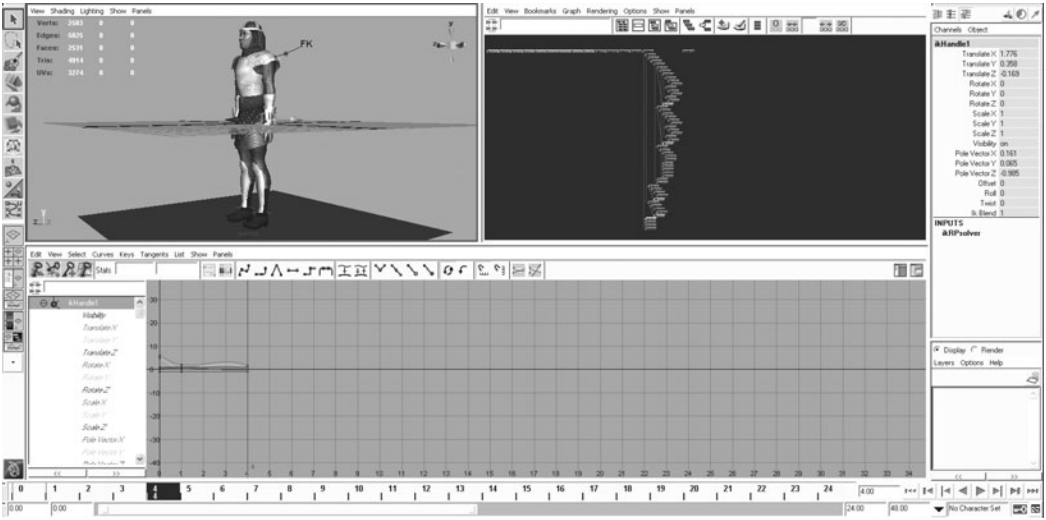


Figure 8.65
Bring the character's arms straight down by his sides.

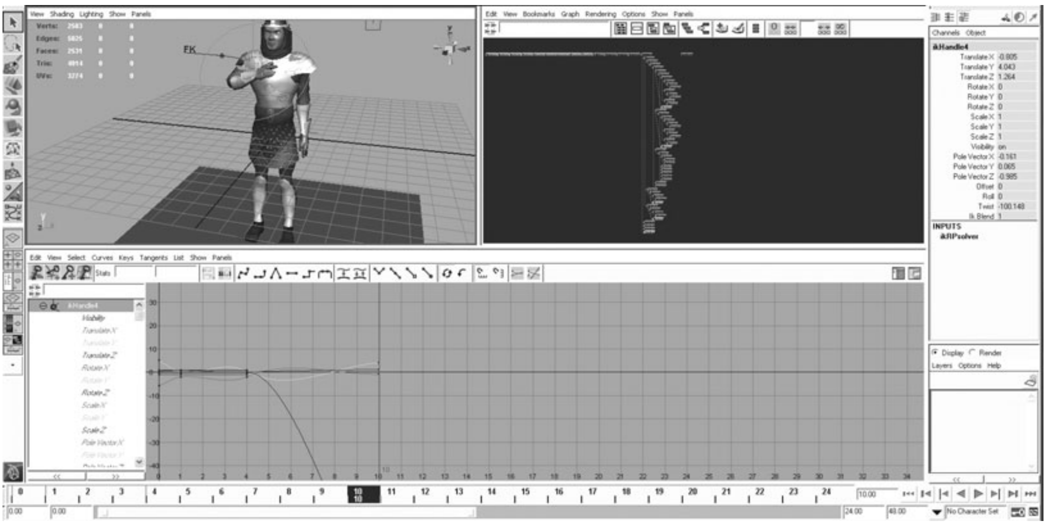


Figure 8.66
Rotate the elbow out.

5. Close the fingers of the hand to form a fist, as shown in Figure 8.67.
6. At frame 6, move the IK handle for the wrist out and away from the character so it does not pass through the chest when raised, as shown in Figure 8.68.

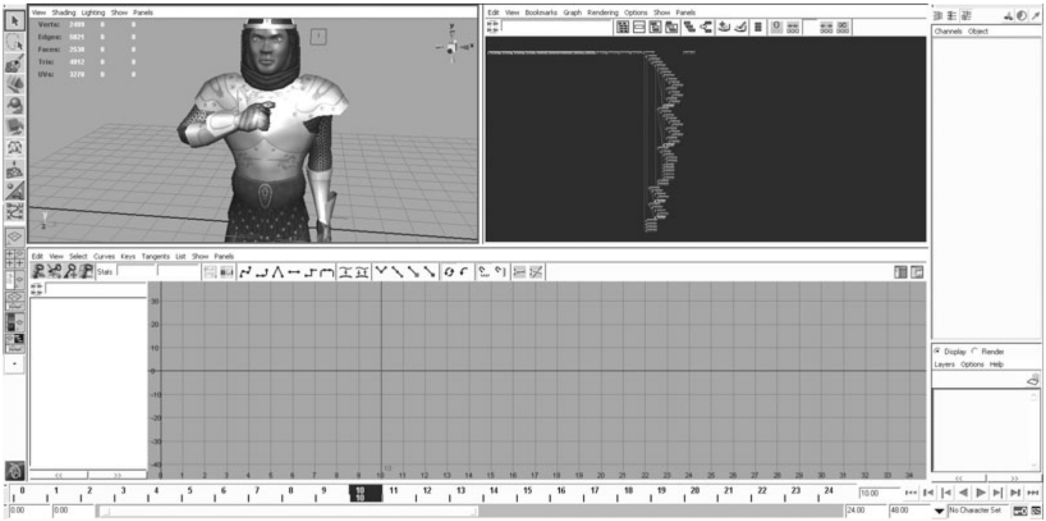


Figure 8.67
Have the hand make a fist in salute.

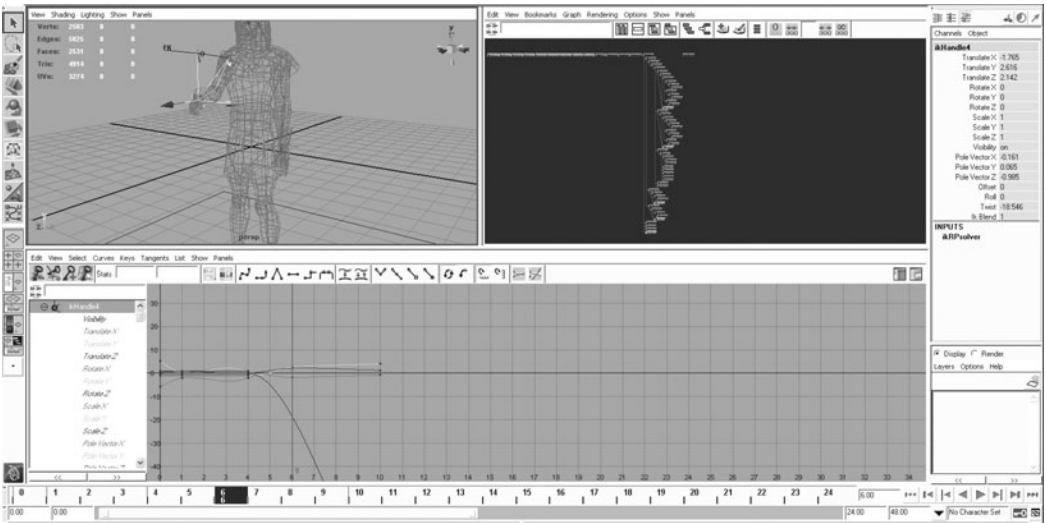


Figure 8.68
Move the hand out at frame 6.

7. Notice that many of the lines in the Graph editor are starting to loop quite erratically. As a result, the character will not make smooth transition from one frame to another. In some cases, these looping actions are desirable but not in a rigid salute animation. To understand this, go to the Graph editor and select the nodes in frame 1; you will notice that all the tangents for that

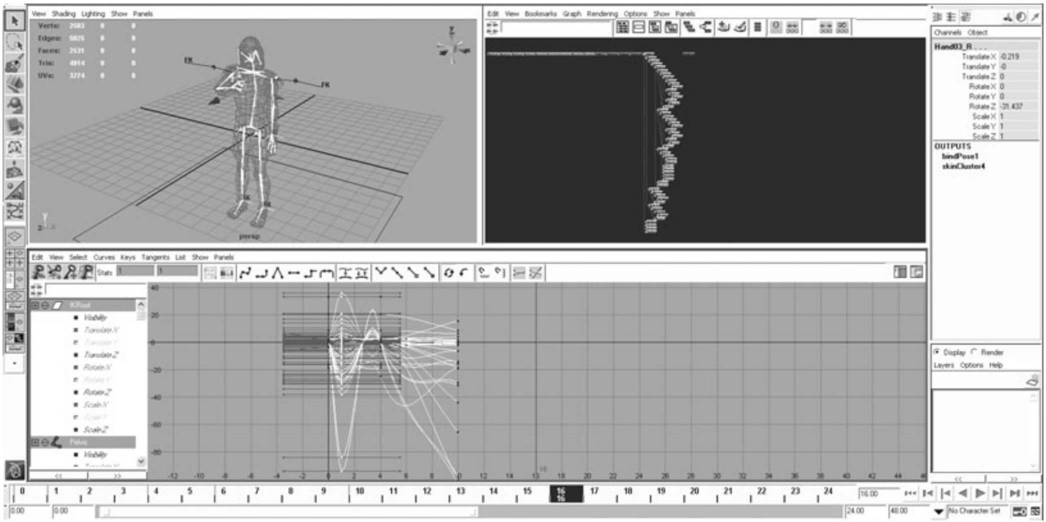


Figure 8.69
The tangents at frame 1 are flat.

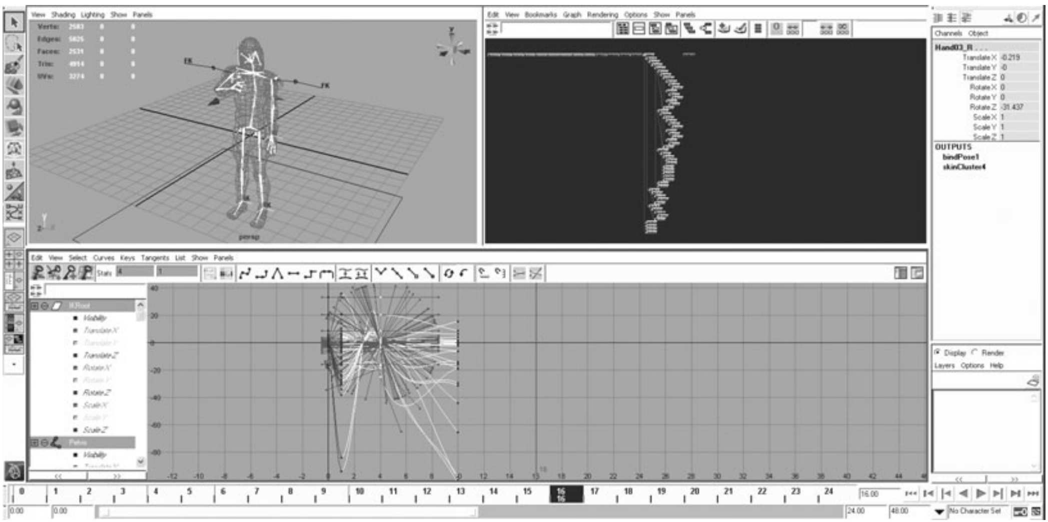


Figure 8.70
The tangents at frame 4 have many angles.

frame are flat (see Figure 8.69). If, however, you select the nodes at frame 4, you will notice that the tangents come in at many angles, as shown in Figure 8.70. On the shelf just above the Graph editor are several tools for changing the attributes of the tangents of a keyframe; dark blue lines with dots on

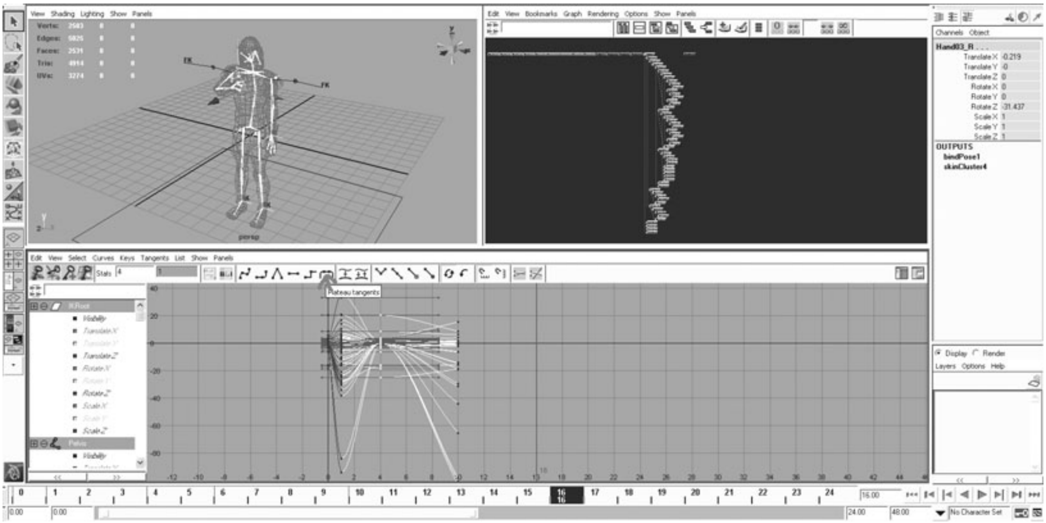


Figure 8.71
Plateau the tangents at frame 4.

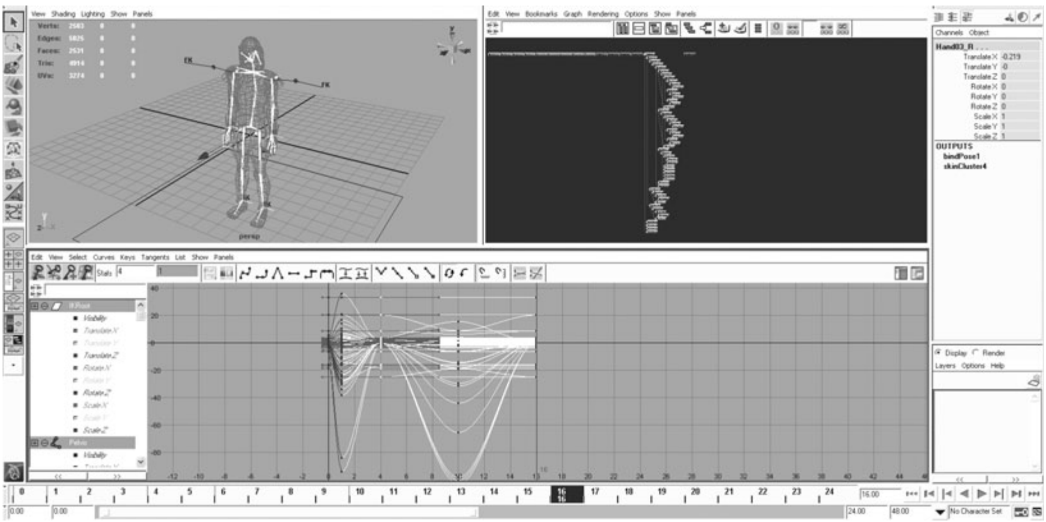


Figure 8.72
Return the character to his attention pose at frame 16.

them represent their function. With nodes on frame 4 selected, click the Plateau Tangents button (see Figure 8.71) to eliminate the looping.

8. Copy frame 4 to frame 16. Remember to select all joints when you copy and paste. This returns the character to the attention pose, as shown in Figure 8.72.

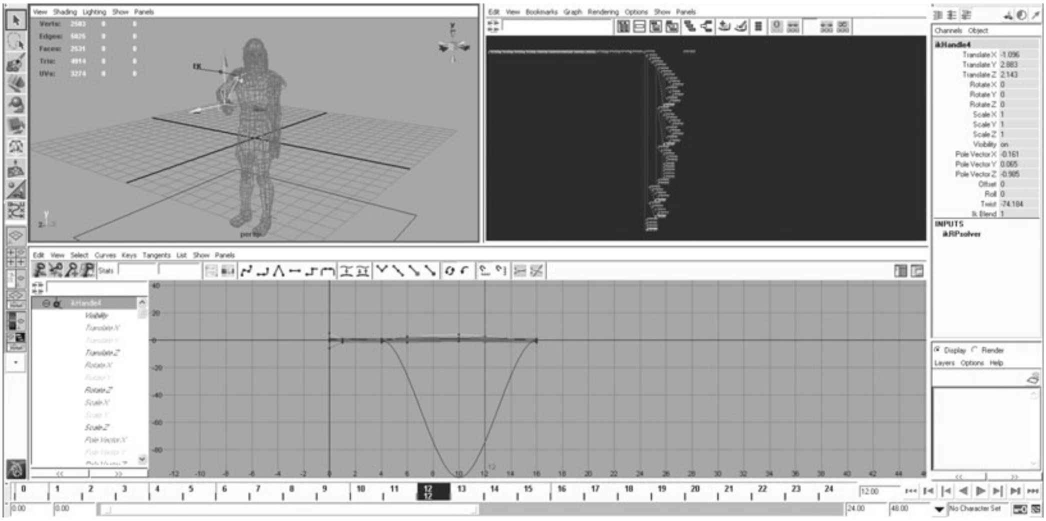


Figure 8.73
Make sure the hand does not pass through the chest on frame 12.

9. Move the hand out away from the chest at frame 12 so it doesn't pass through the chest, as shown in Figure 8.73.
10. Copy frame 1 to frame 20 to move the character back to his relaxed pose.
11. Notice that there is a slight jump in the motion of the arms as the character goes from the attention pose to the relaxed pose. This again is a looping motion problem. Select the IK handle for one of the wrists and look at the motion lines in the Graph editor. You will see where a couple of them go up past the node and then back down again. If you click one of the problem nodes, you will see the tangent handles appear. If you have the Move tool selected, you can select one of the handles and move it using the middle mouse button (see Figure 8.74).

There you go. You just completed a simple character animation in Maya. Play the animation to see how it looks. Although the animation my not be completely lifelike, it does get in all the major motions. To get truly lifelike motion you will need to adjust and fine-tune the animation extensively, but this is a good start. Congratulations!

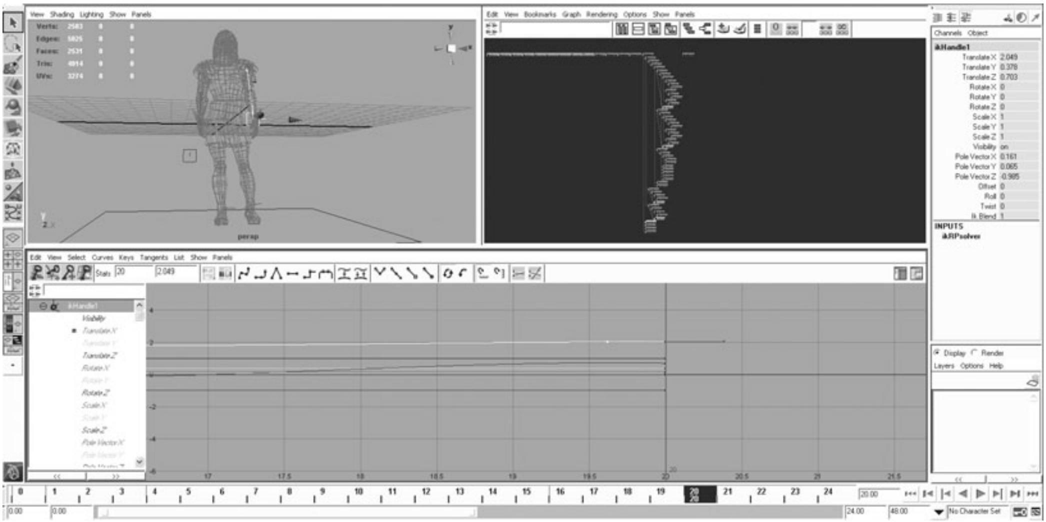


Figure 8.74
Fix the problem nodes by adjusting the tangent handles in the Graph editor.

Summary

This chapter covers a lot of information on animating game characters in both 2D and 3D applications. I hope you were able to follow along with the exercises and feel comfortable with the tools for animating.

In this chapter you learned the following:

- The definition of animation
- How animation is drawn
- How to squash and stretch in animation
- Weight and balance in animation
- Animation arcs
- Animation cycles
- Similarities between frames in a walk animation
- How to create a walk cycle
- Maya's animation controls

- Moving objects
- Path animation
- Creating a simple character animation
- Using the Graph editor to adjust the animation

In the next chapter you will learn how to animate a more complex animation for your 3D character. Before you move on, spend a little time exploring some of the other animation tools in Maya's Graph editor and animation controls.

This page intentionally left blank

Chapter 9

Animation Cycles

In game animation, most of the animation work you will do will involve cycles. An *animation cycle* is an animation that starts and stops on the same frame so that it can be repeated, or *looped*. This type of animation works well for repetitive actions like walking, where there only needs to be one animation of the action of moving both feet through the motion. The animation can then be repeated again and again, and the character can walk forever. Other actions, such as swinging a sword or firing a gun, can also be considered cycles. Even though they may not continuously repeat, they *are* repetitive actions that a character may perform a number of times.

By starting and ending an animation on the same frame, you allow the game character to act in the world with a minimum of actual frames of animation, while still seeming to be active. For example, a typical set of animation cycles for an enemy monster in an action game might be as follows:

- Walk
- Run
- Attack
- Pain
- Death
- Idle

With just these six animation cycles the monster can move about the environment, attack the character, stand looking around, and die. What more do you need from a monster?

In this chapter you will learn how to create a walking animation cycle for your 3D character. The walking animation is probably the most viewed animation of all animations and, therefore, is the most critical to get right. There are a lot of different ways to walk. Some people walk boldly, while others are timid. Some people walk with a bounce, while others try to be smooth. There are almost as many ways to walk as there are people on this planet. Here, you will work on a basic walk that looks normal for a game character, which you will be able to use on all of your characters. After you get the basic walk down, you can experiment with different styles to add personality to your characters.

Moving the Feet

Walking is mostly about moving the body from one place to another. To accomplish that movement, we use our legs and feet. It makes sense, then, to start a walk animation by moving the character's feet.

1. Load your character with the bind pose(the beginning pose you created in the last chapter.)

2. Go to the side view and lift the character's left foot in frame 1, as shown in Figure 9.1.

3. Start the walk animation on a frame that is close to the normal standing frame; that way, there won't be a big jump in the transition between standing and walking. To begin, select all the nodes in Hypergraph and set a keyframe to lock them in place as shown in Figure 9.2. You can adjust them later, but it is a good idea to have all the attributes set as a keyframe on the first frame of animation.

4. Move the Current Frame indicator to frame 8.

5. Select the pelvis joint and lower it slightly to give your character a little bounce in his walk. You can track the movement in the Channel box and in the Graph editor. Your character should bend at the knees as shown in Figure 9.3.

6. When we step forward while walking, our body naturally comes down; in contrast, when we pull our feet together, our body naturally rises. This

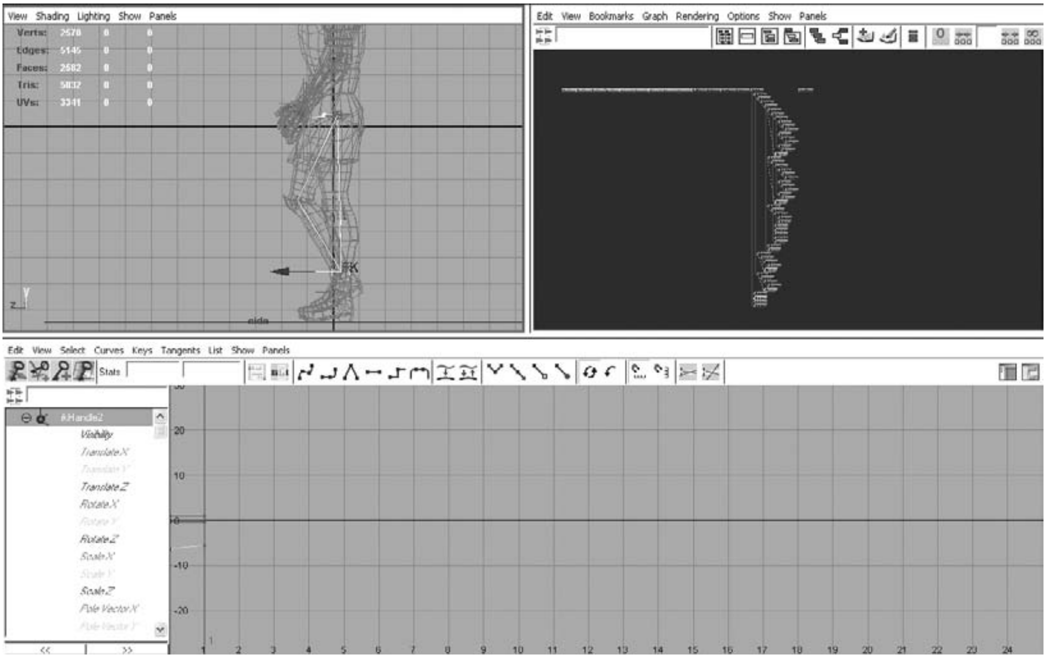


Figure 9.1
Use the IK handle to lift the character's left foot.

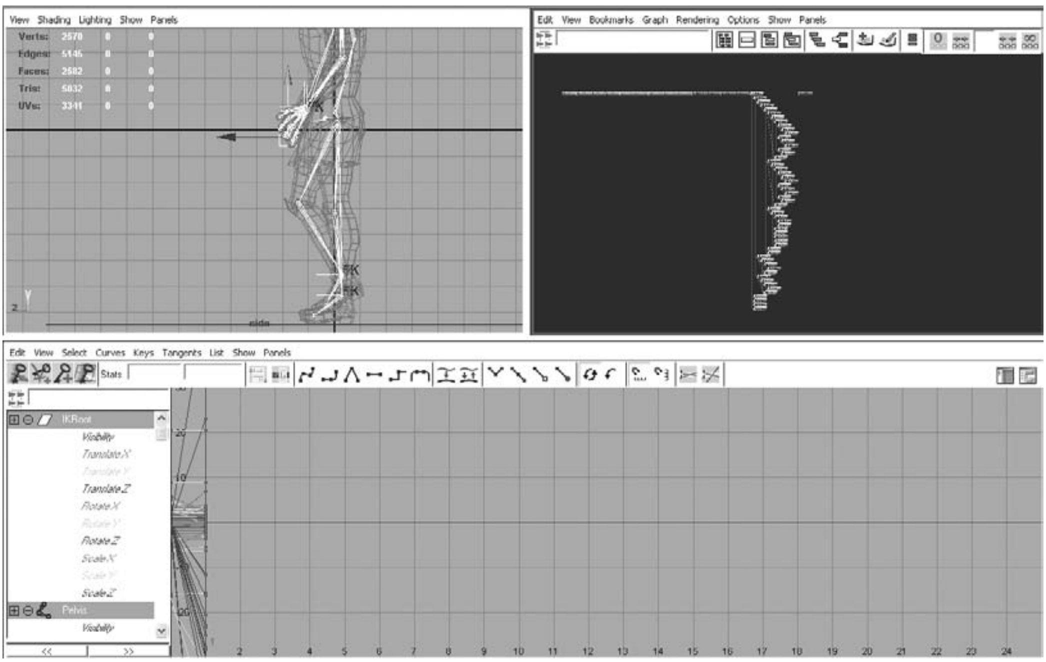


Figure 9.2
Set a keyframe at frame 1.

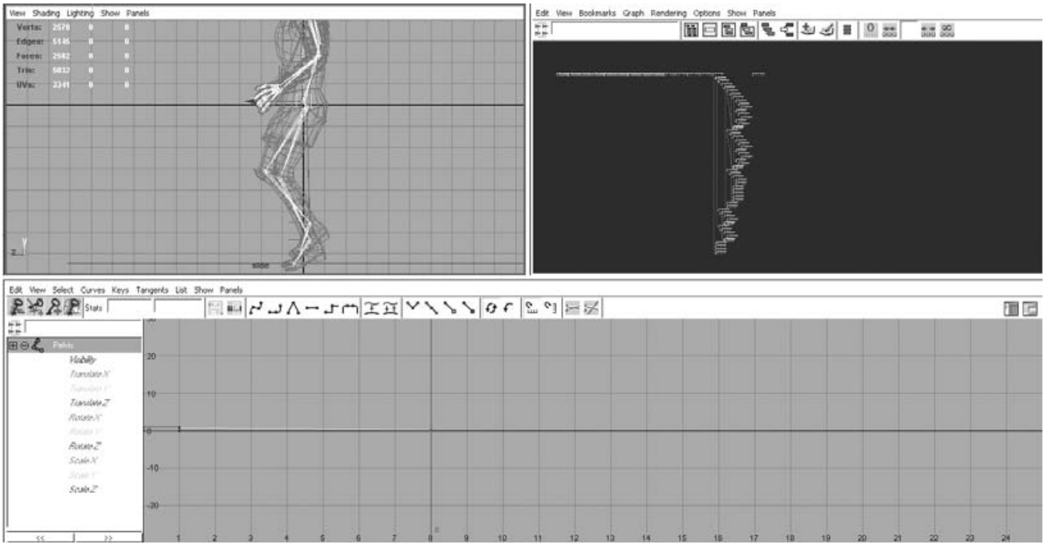


Figure 9.3
Pull the pelvis down to have your character bend at the knees.

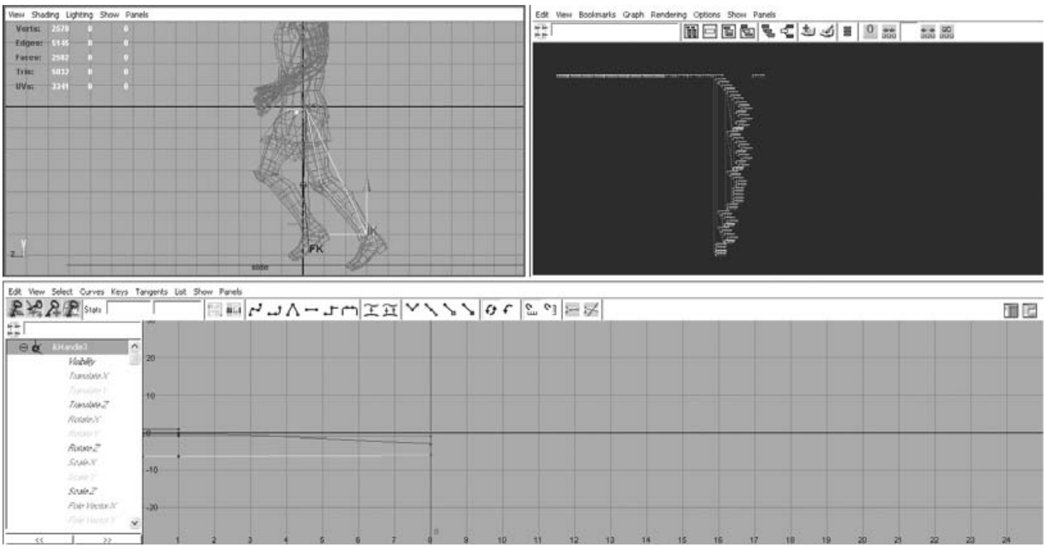


Figure 9.4
Use the IK handle to pull the right foot back.

motion should be smooth and even. To reflect this, pull the character's right foot back as shown in Figure 9.4.

7. Move the character's left foot forward to complete the open stride, as shown in Figure 9.5.

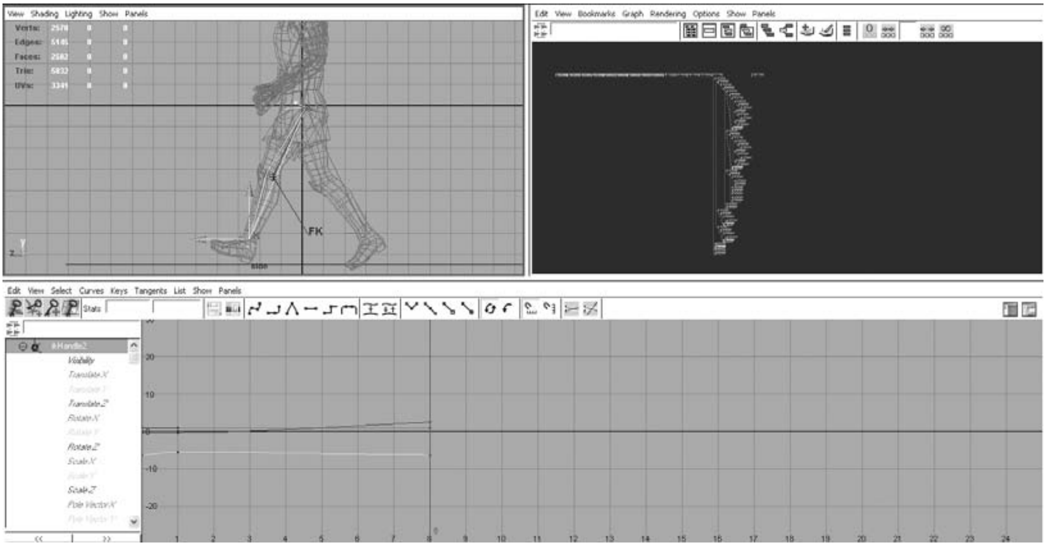


Figure 9.5
Use the IK handle to move the left foot forward.

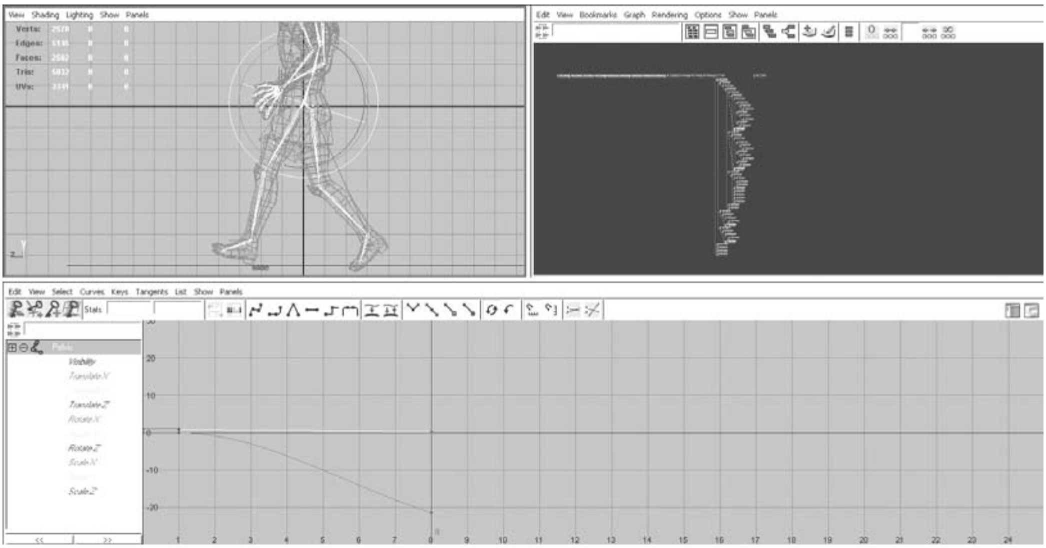


Figure 9.6
Rotate the pelvis to get a swinging action in the hips.

8. When a person walks, his hips move with the motion of the legs. To accommodate this, select the pelvis joint and rotate it about −21 degrees in the x axis, as shown in Figure 9.6.

9. In a walking cycle, the two sides of the body mirror each other in action but are offset by a half step. That means that the left side uses exactly the same

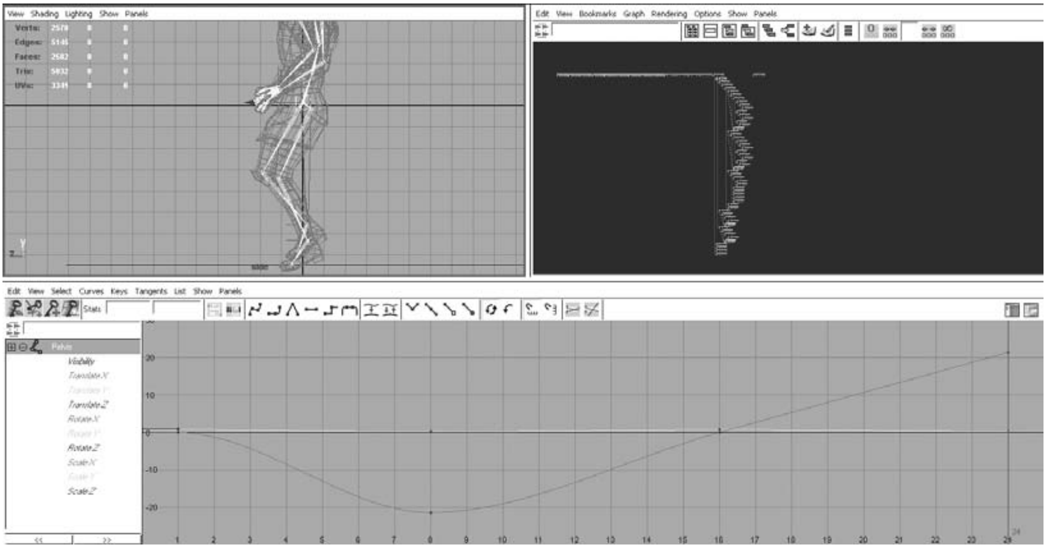


Figure 9.7
Frame 1 and frame 16 are similar.

motion as the right side, except that it does it a half step later. To effect this, move to frame 16. This frame needs to be the same as frame 1 except that right foot needs to be lifted instead of the left (see Figure 9.7).

10. Frame 24 is the same as frame 8, but mirrored so the right leg is forward instead of the left foot, as shown in Figure 9.8.

11. Complete the motion of the feet by repeating frame 1 at frame 32. (You will need to extend the animation sequence to 32 frames, as shown in Figure 9.9.)

You have just finished the basic movement of the feet for a walk cycle; play the animation back to see the legs move. If you set up the movement correctly the feet should start and stop at the same position. The body of the character should also move up and down with each stride. The animation is far from finished because the action is still very stiff, but you are off to a good start.

Note

As you worked on your walk animation, you may have noticed the motion lines in the Graph editor. In a walk cycle, the motion lines tend to make a sweeping S curve from start to finish. You will see this S curve repeated throughout this animation. Depending on joint the curve may be red, green, or blue. A blue curve means that the joint is rotating in its z axis. A red curve means that the joint is rotating in its x axis. A green curve means that the joint is rotating in its y axis. In the case of the IK handles, the curves are for translation rather than rotation. You can use the colors to help you understand what is happening in your animation and tune the animation as needed.

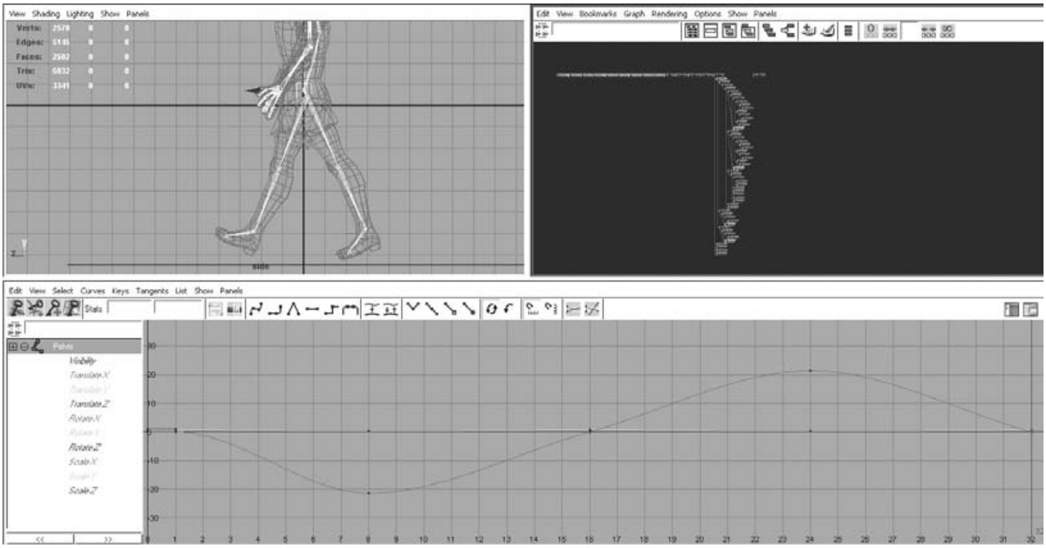


Figure 9.8
Mirror the action of the legs in frame 24 from frame 8.

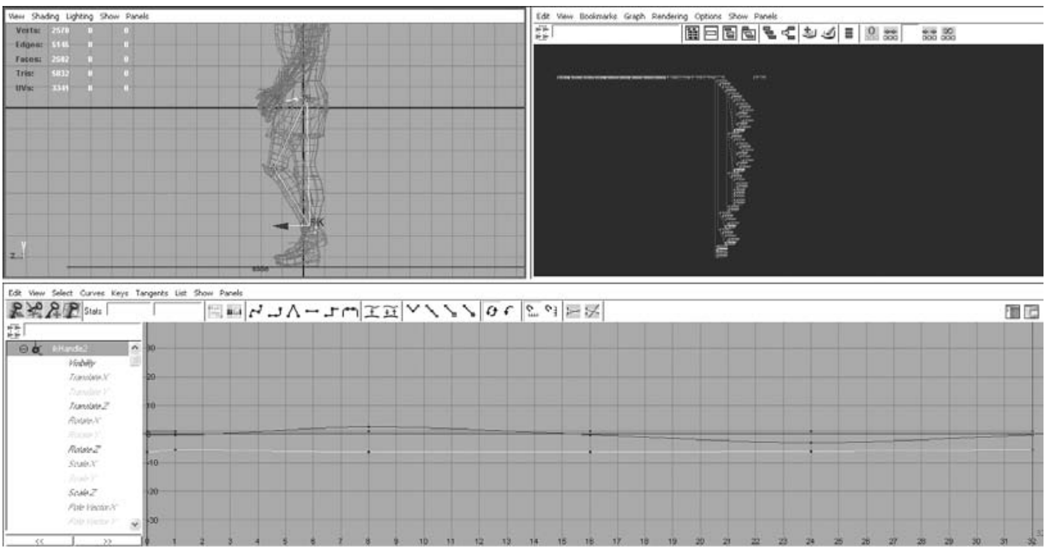


Figure 9.9
Frame 1 and frame 32 must match to complete the cycle.

Arm and Chest Movements

Try walking with arms coming forward with your legs. Do you feel how uncomfortable it is? It is unnatural for the body to have the arms swing with the legs. When a person walks, the arms move counter to the legs. The chest also

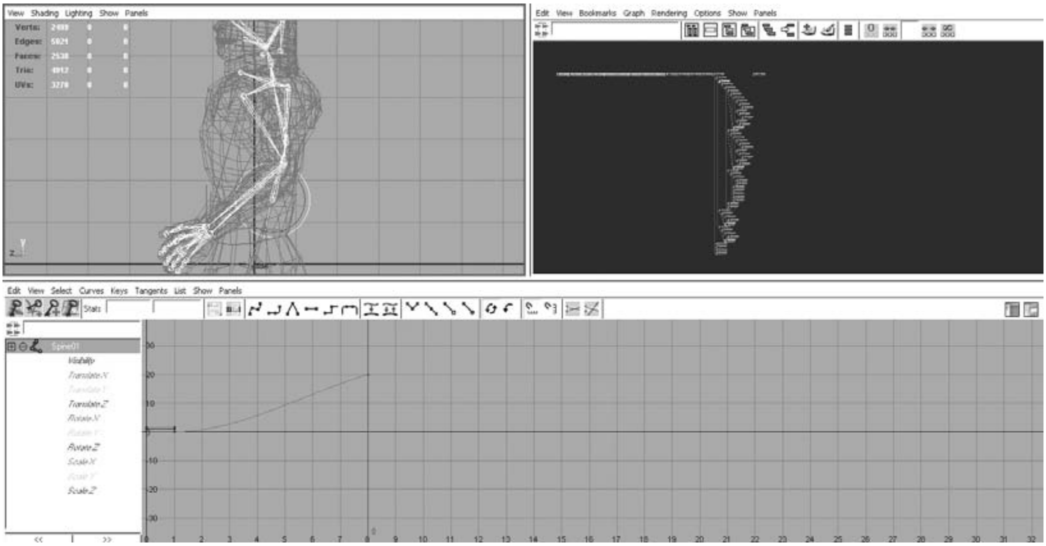


Figure 9.10
Rotate Spine1 20 degrees in the x axis.

twists opposite the hips. This natural rhythmic counteraction is designed to help keep the body balanced during a walk. To get the natural swing of the character's arms and chest, you first need to work with the spine joints to twist the upper torso counter to the hips.

1. Select the Spine1 joint and rotate is 20 degrees in its x axis as shown in Figure 9.10 at frame 8.

2. Now go to Spine2 and rotate it 10 degrees in the x axis, as shown in Figure 9.11.

3. Spine3 has a different orientation than Spine1 and Spine2. Instead of rotating in the x axis, you will need to rotate it in the y axis. Rotate Spine3 in the y axis 10 degrees as shown in Figure 9.12.

4. The character's upper body is rotating opposite of the lower body as the character makes a stride in frame 8. Next, go to frame 16 and set the rotations of all of the spine joints to 0, as shown in Figure 9.13.

5. At frame 24, the chest needs to swing opposite the legs again. Rotate the spine joints in a manner similar to how you did at frame 8, except that instead of rotating in the positive direction, rotate them in the negative

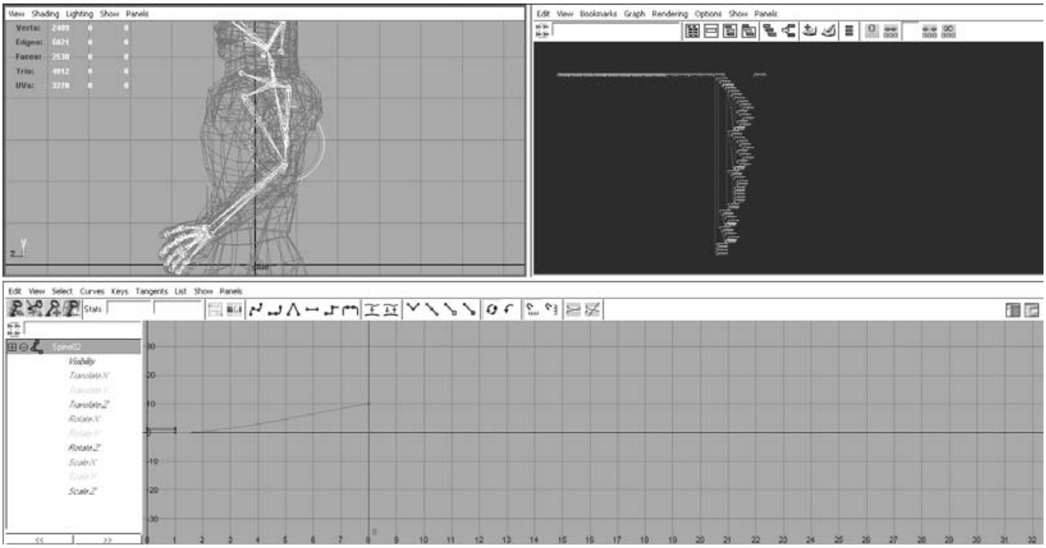


Figure 9.11
Rotate Spine2 10 degrees in the x axis.

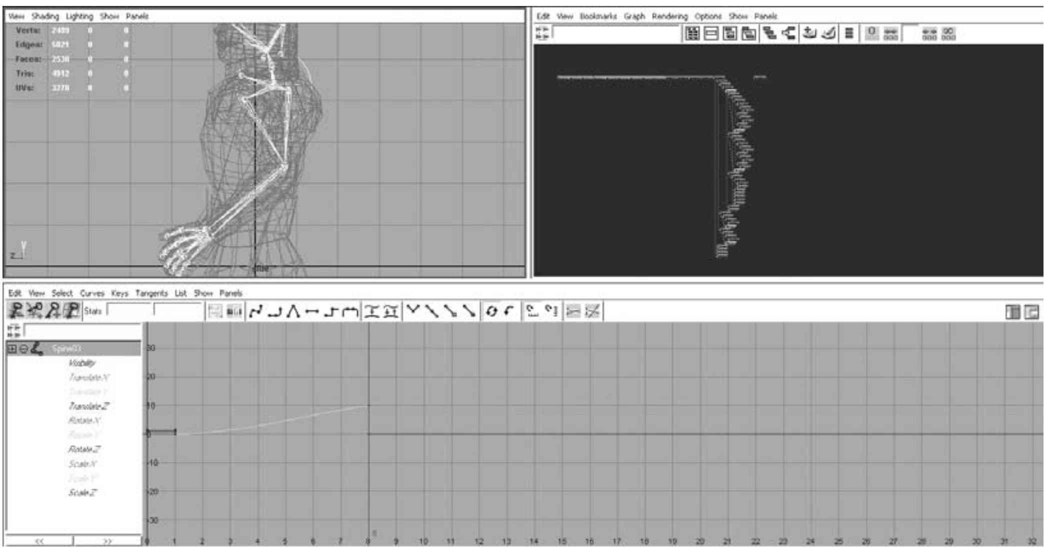


Figure 9.12
Rotate Spine3 10 degrees in the y axis.

direction. Rotate Spine1 −20 in the x axis, Rotate Spine2 −10 in the x axis, and rotate Spine3 −10 in the y axis as shown in Figure 9.14.

6. To complete the swinging of the chest, go to frame 32 and rotate all the joints back to 0, as shown in Figure 9.15.

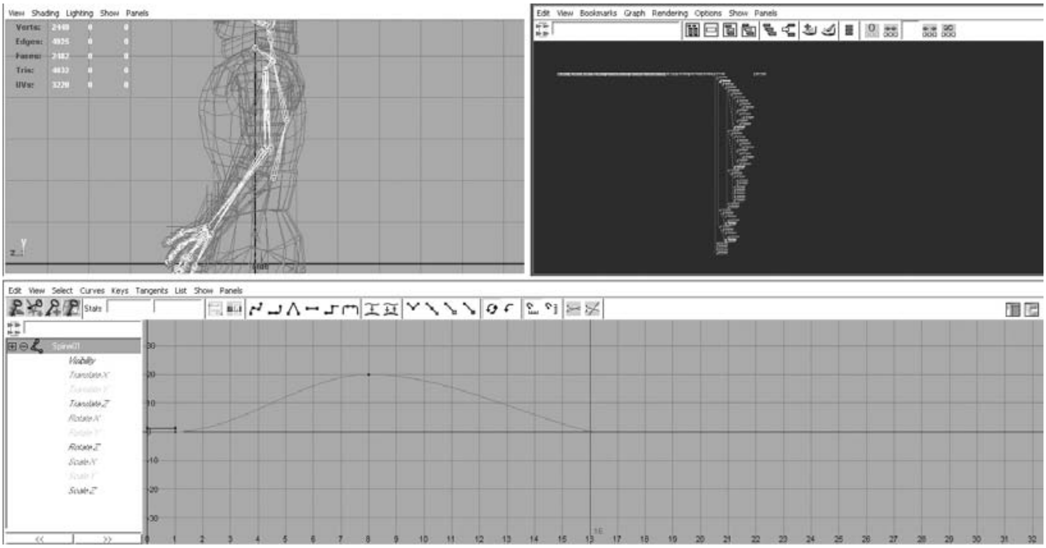


Figure 9.13
Rotate the spine joints back to 0 at frame 16.

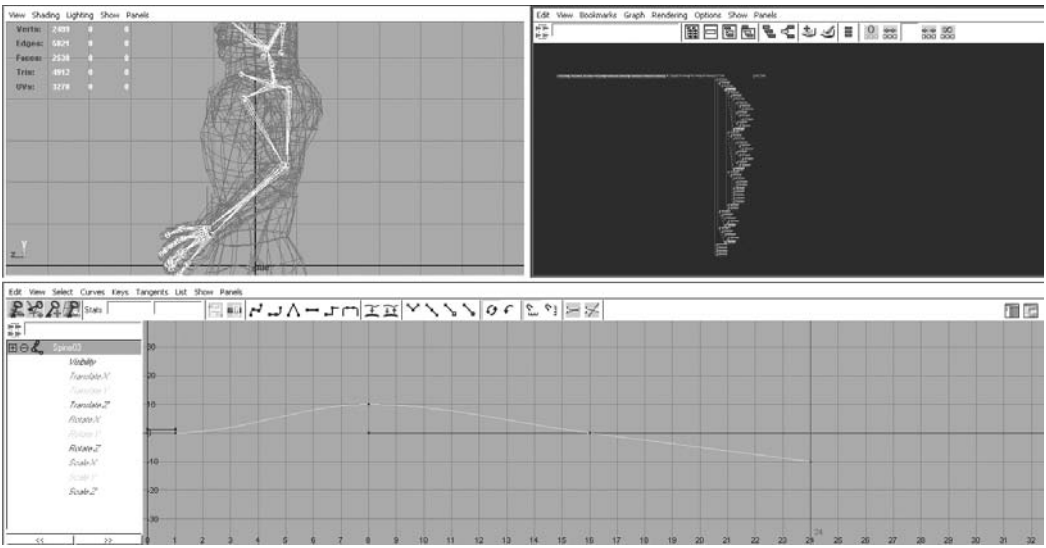


Figure 9.14
Have the chest swing counter to the legs at frame 16.

7. Next, move the character's arms. Begin in frame 8, moving the character's right arm forward and his left arm back as shown in Figure 9.16.

8. In frame 16, move the arms back to the same position as in frame 1.

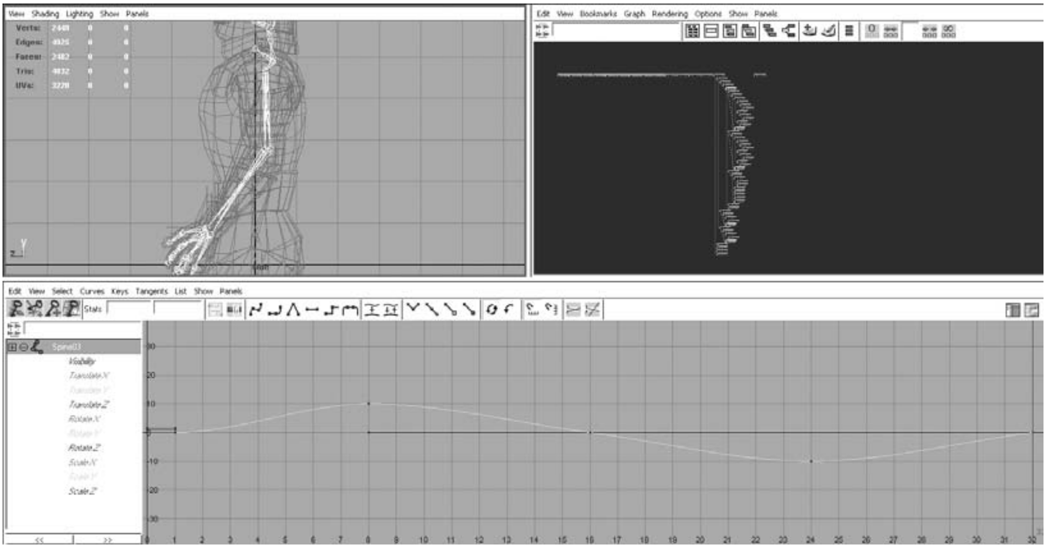


Figure 9.15
Rotate the spine joints back to 0 at frame 32.

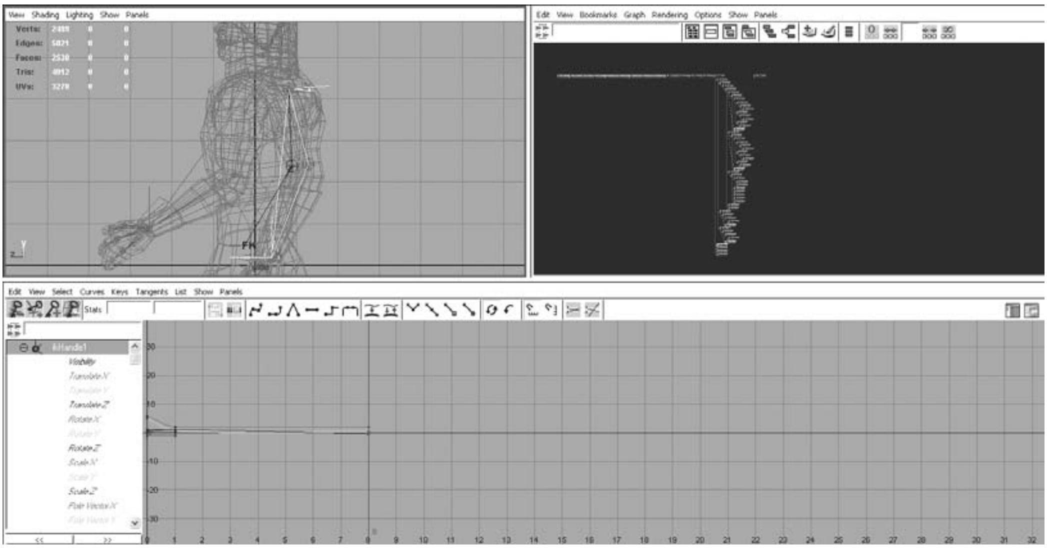


Figure 9.16
Move the arms counter to the legs at frame 8.

Note

You can use the Channel box to help you match frames. Notice that there are translation and rotation numbers for the IK handles and for the joints in the Channel box; if these numbers match exactly between two frames, the frames will be identical.

9. Swing the arms opposite the legs in frame 16.
10. Bring the arms back to the character's sides and match frame 1 with frame 32 to complete the cycle.

The arms and the legs are now swinging on the character counter to each other and your animation is progressing. You still have a way to go to complete your animation, but you have the major motion working. You will probably notice that your character still looks somewhat robotic, however; that is because you still need to make several adjustments to the animation to get it to look right.

Animating the Feet

As you look at your walk animation, you will see that the feet are just swinging with the legs as opposed to reacting to the ground. In fact, they pass *through* the ground in a number of places instead of serving as the contact point between the body and the ground. The feet play a vital part in a walk cycle; you must animate them so they appear to be striking the ground and pushing off it as the character walks.

1. In frame 8, where the back foot should be bent pushing off of the ground, rotate the ankle and foot joints to bend the foot as shown in Figure 9.17.

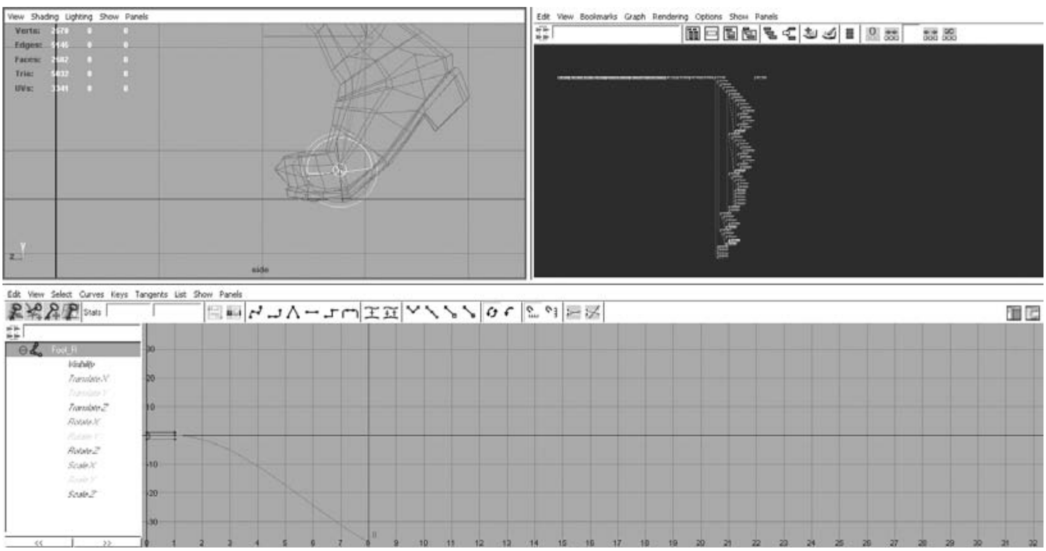


Figure 9.17
The back foot should be pushing against the ground at frame 8.

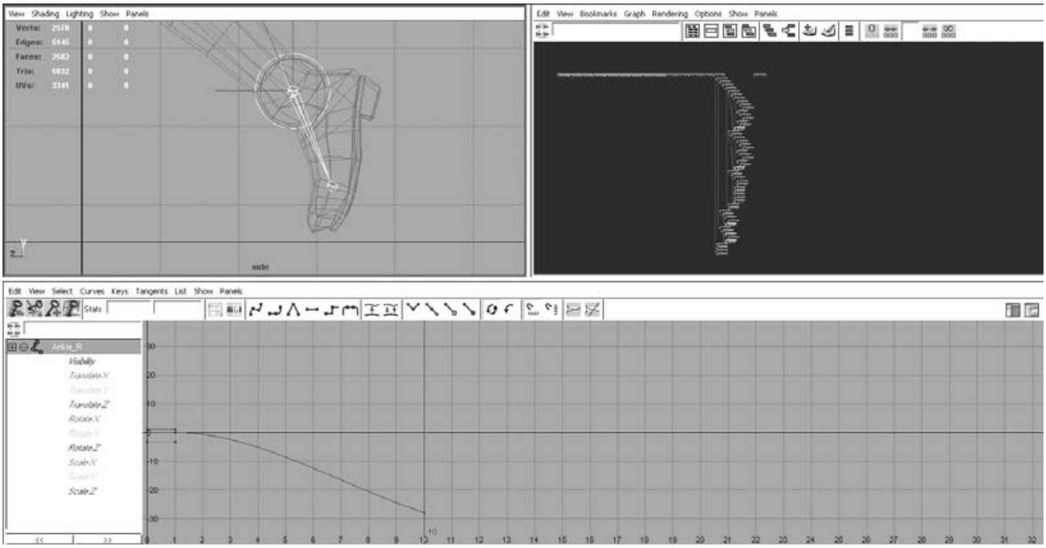


Figure 9.18
The foot should flip back at frame 10.

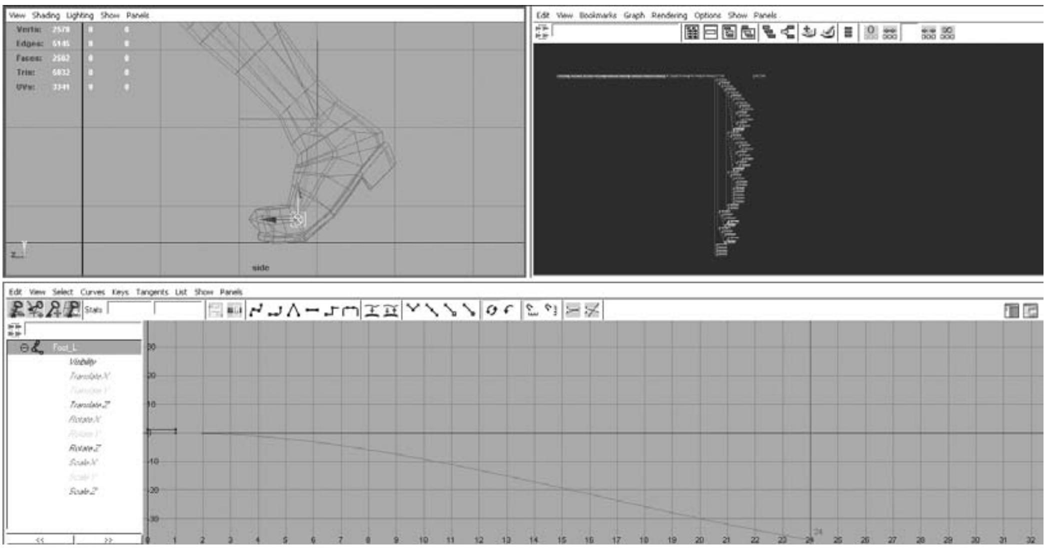


Figure 9.19
The other foot should also bend and flip when it comes to the back of its swing.

2. As the back foot comes off the ground, it should flip back as it propels the body forward. This flip action should happen between frame 8 and frame 10. Go to frame 10 and flip the foot back as shown in Figure 9.18.

3. Now you can animate the compression and flip of the other foot at frame 16 as shown in Figure 9.19.

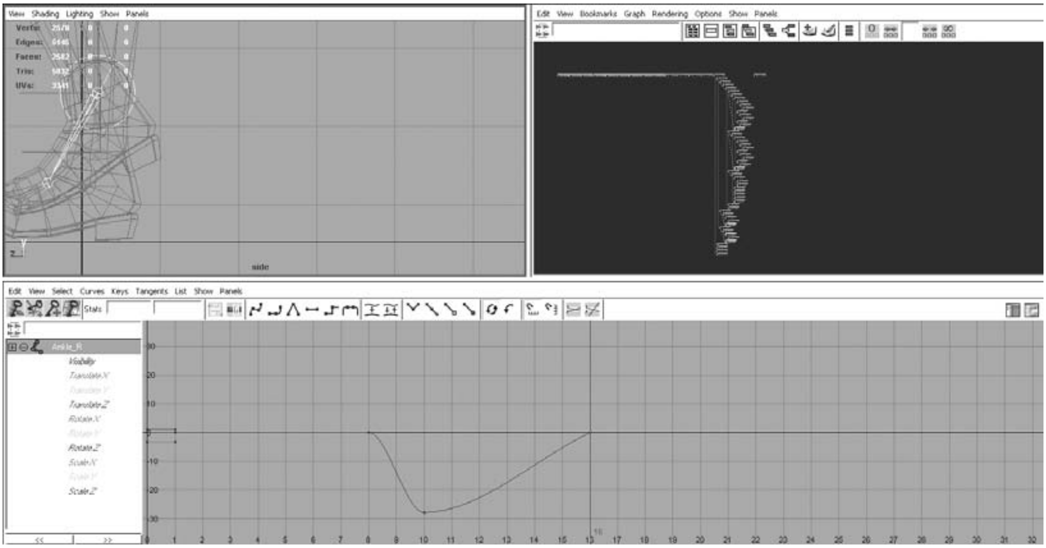


Figure 9.20
Rotate the right foot forward at frame 16.

4. Go to frame 16 and notice that the backward flip of the right foot is still present and needs to be adjusted so it is not dragging as it comes forward. To do so, bring the rotation of the ankle joint back to 0 as shown in Figure 9.20.
5. At frame 32, rotate the left ankle joint back to 0.
6. Now you need to animate the foot coming into contact with the ground. This involves a rolling motion, with the foot striking the ground first with the heel and later with the ball of the foot. To begin, go to frame 8, where the heel is beginning to come in contact with the ground. Rotate the ankle to bring up the front of the foot as shown in Figure 9.21.
7. Go to frame 10 and plant the foot as shown in Figure 9.22. Notice the double S curve of the ankle joint along the z axis in The Graph editor.
8. Animate the planting of the right foot at frame 24 and frame 26 as shown in Figure 9.23.
9. You will notice that the rotation of the ankle looks odd now at frame 16. To fix it, go to frame 16 and return the ankle rotation to 0 as shown in Figure 9.24.
10. There is a similar problem for the ankle rotation of the left foot at frame 16. To fix it, rotate the left ankle joint back to 0 on frame 16, as shown in Figure 9.25.

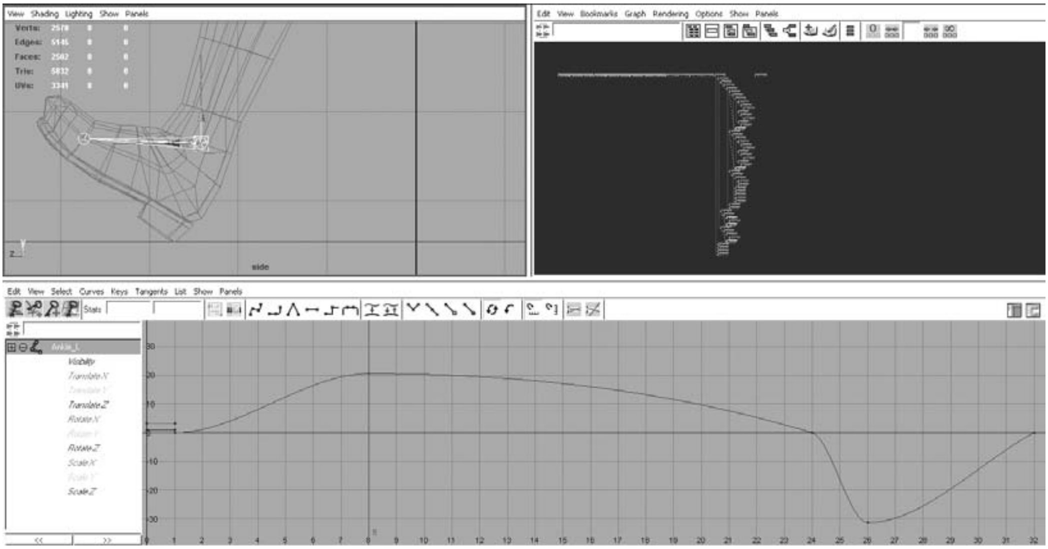


Figure 9.21
Bring the front of the left foot up at frame 8.

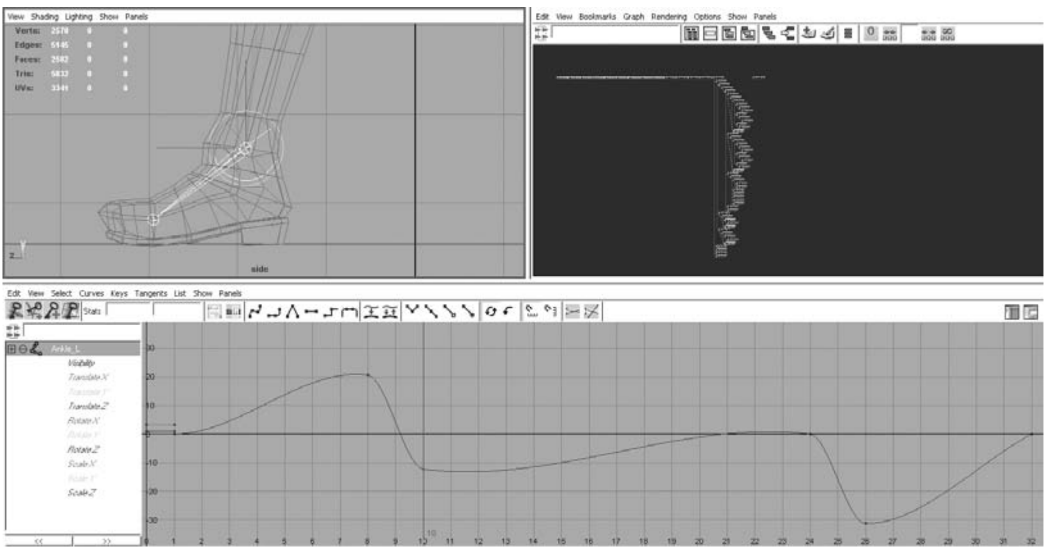


Figure 9.22
Plant the foot on the ground at frame 10.

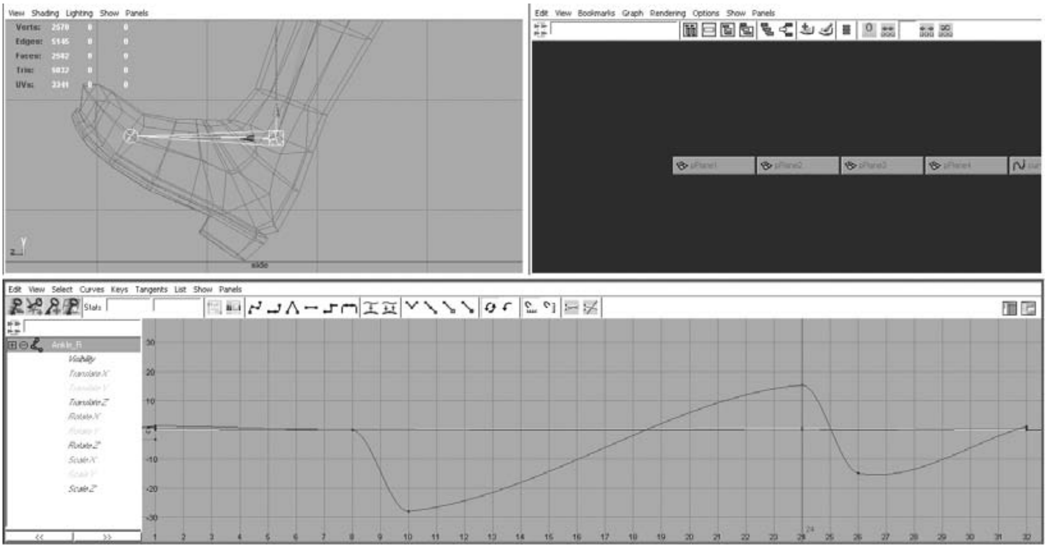


Figure 9.23
Plant the right foot at the end of its forward swing.

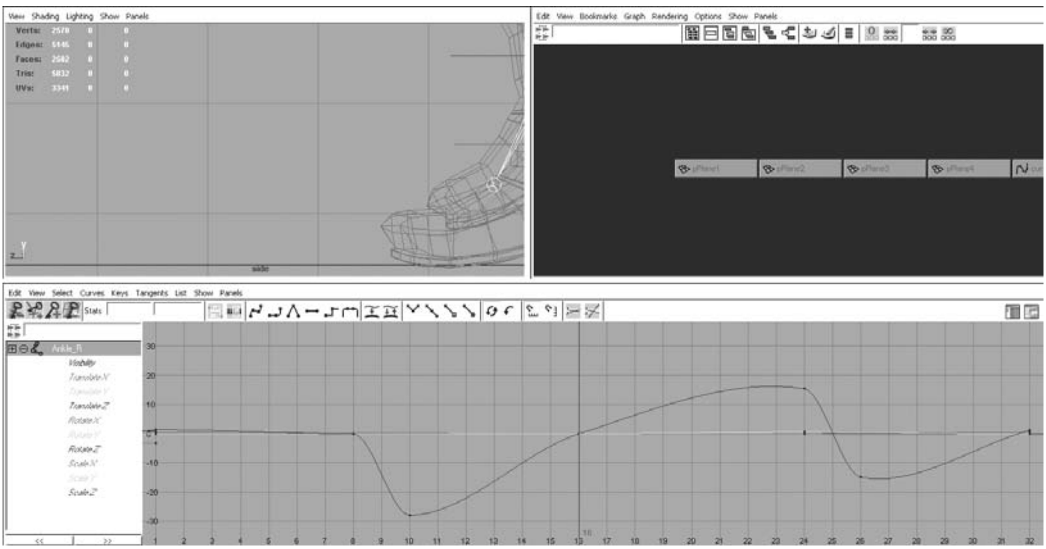


Figure 9.24
Fix the rotation problem at frame 16.

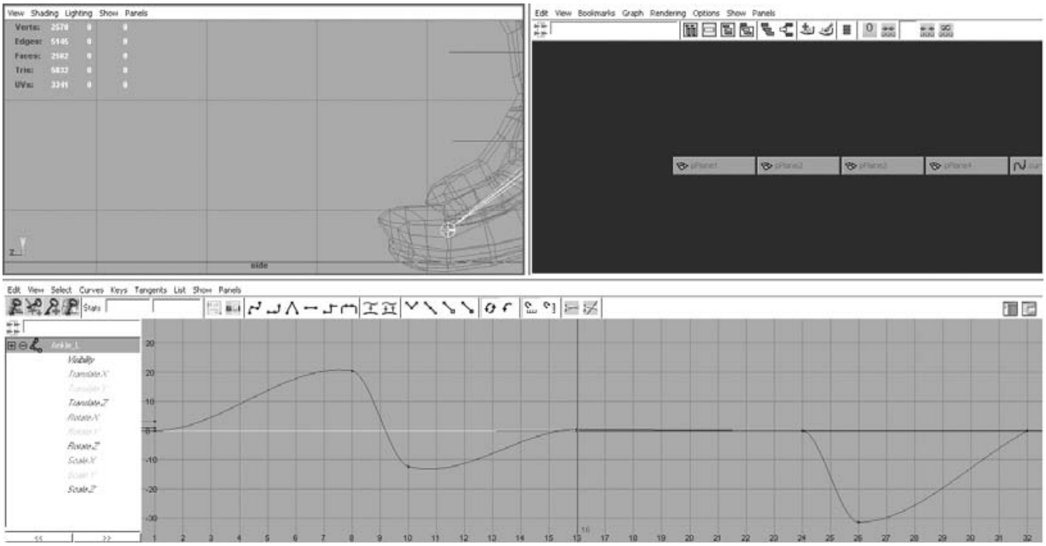


Figure 9.25
Rotate the left ankle back to 0 on frame 16.

Play the animation and see how the feet look. They should look as if they are pushing off of the ground as they go back and like they are planting themselves on the ground at the end of their forward motion.

3D Movement

The animation is starting to look a little better from the side, but, this is not a 2D animation. The character will be seen from *all* sides. That means it has to look right from any angle. Start with the hips. When someone walks, a natural up and down motion of the hips occurs. Likewise, there is a swinging of the pelvis caused by the impact of the forward leg when it comes into contact with the ground. These up-and-down and swinging motions are noticeable in a walk when viewed from the front and even more so from the back. Wearing high-heeled boots or shoes tends to accentuate the motion. Fortunately, you've already accounted for some of this movement in your animation; to handle the rest, switch to the front view and do the following:

1. Go to frame 10 where the left foot is planted on the ground and rotate the left hip up, as shown in Figure 9.26.
2. Go to frame 16 and rotate the pelvis back to 0 as shown in Figure 9.27.

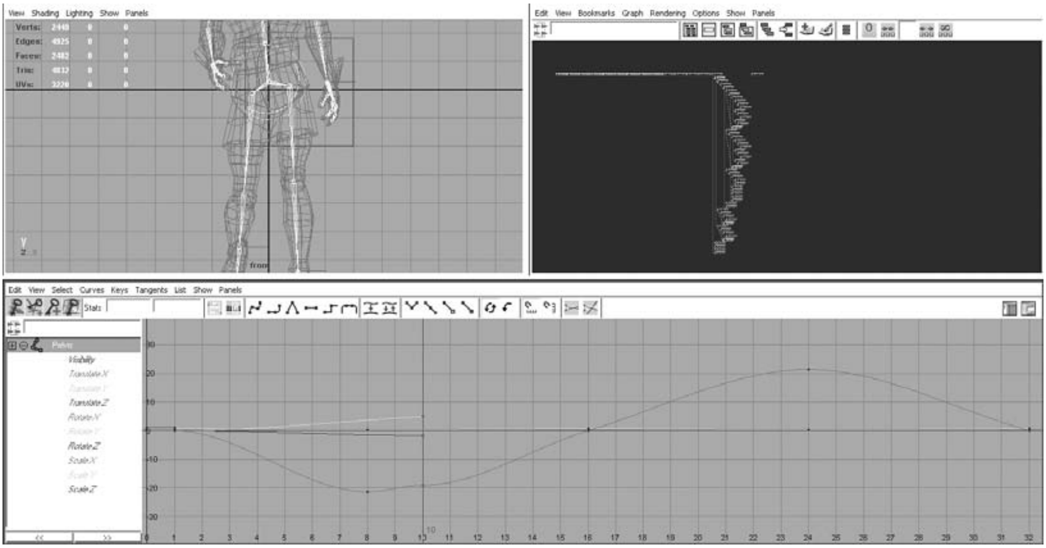


Figure 9.26
Rotate the left hip up at frame 10.

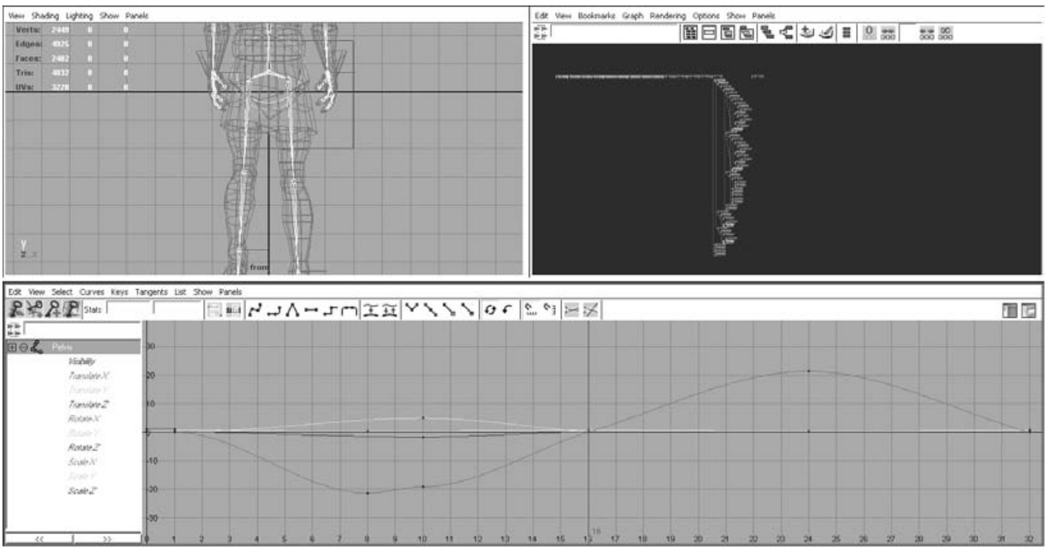


Figure 9.27
The pelvis returns to 0 rotation at frame 16.

3. Go to frame 26 and rotate the right hip up as the right leg is planted on the ground (see Figure 9.28).

4. You probably noticed when you rotated the pelvis that the upper body shifted over with the rotation. To compensate for this, you must rotate the

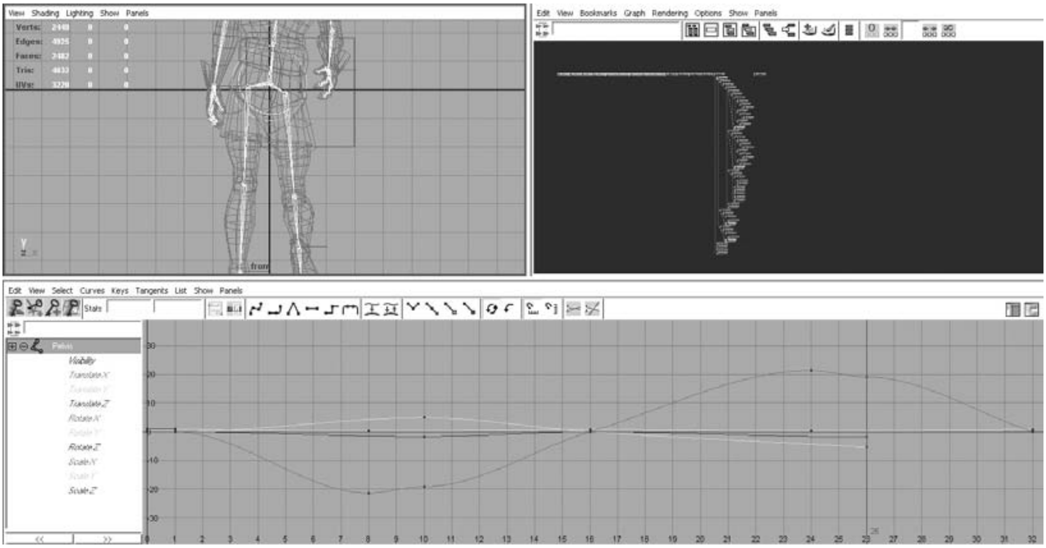


Figure 9.28
The pelvis returns to 0 rotation at frame 16.

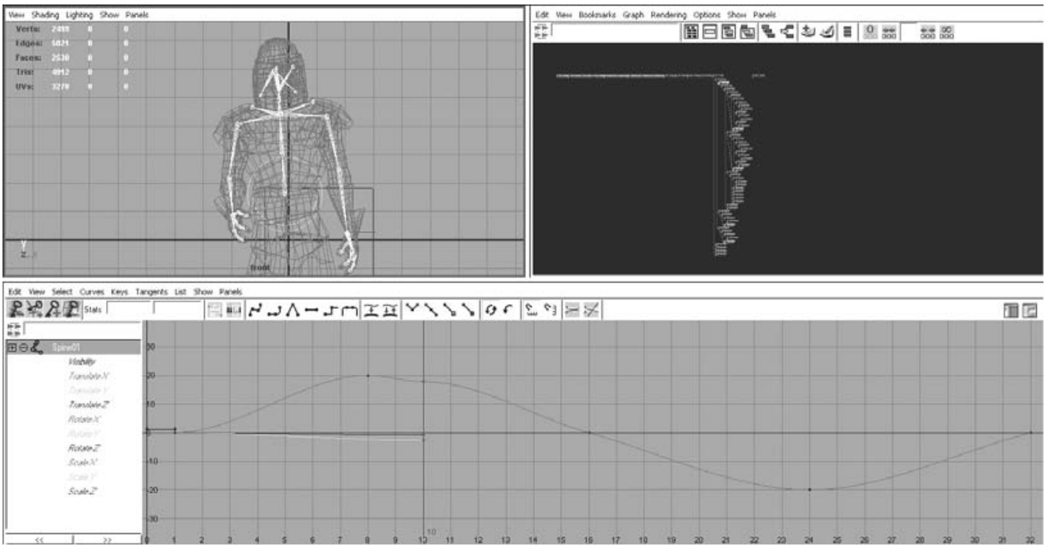


Figure 9.29
Make half of the compensation rotation at Spine1.

spine joints. To do so, go to frame 10 and, at Spine1, rotate the torso about halfway back (see Figure 9.29).

5. Complete the compensation rotation at Spine2, as shown in Figure 9.30.

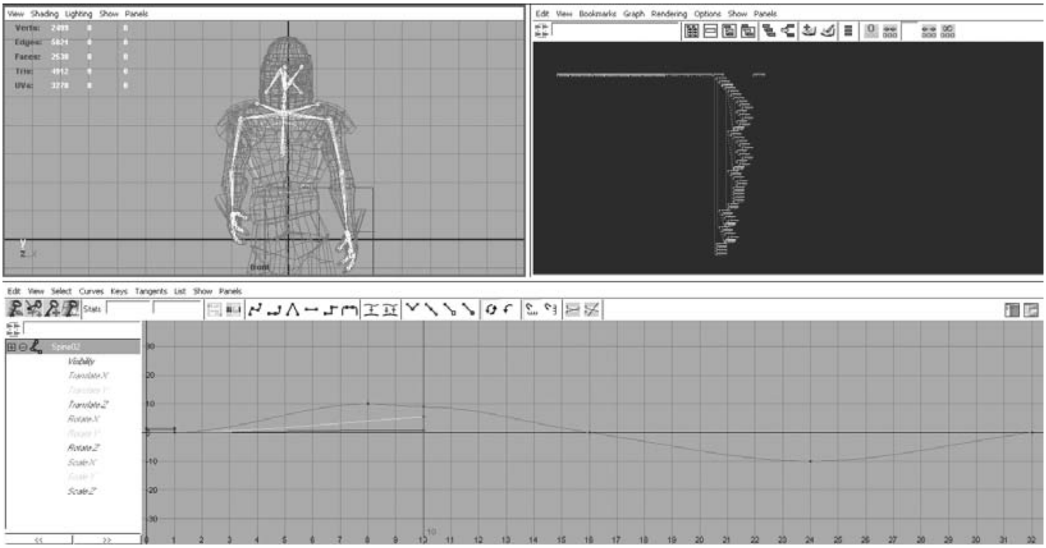


Figure 9.30
Rotate the torso back vertical.

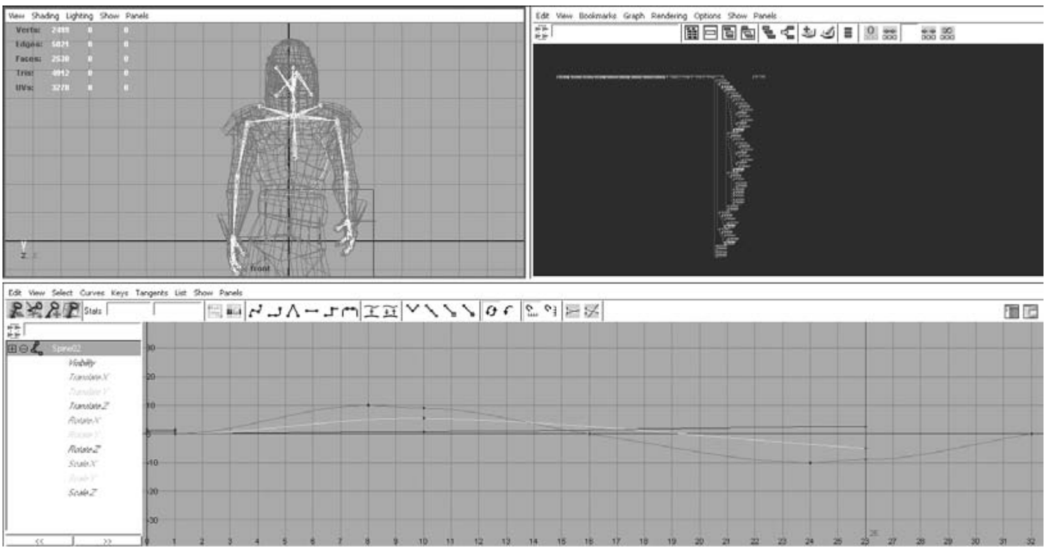


Figure 9.31
Rotate the torso back to vertical at frame 26.

6. Likewise, rotate the torso back to vertical on frame 26, as shown in Figure 9.31.
7. Another side-to-side action of the body when walking is the shifting of balance over the leg that is carrying the weight. The slower the walk, the more pronounced this shifting becomes; in a fast run cycle, there is almost

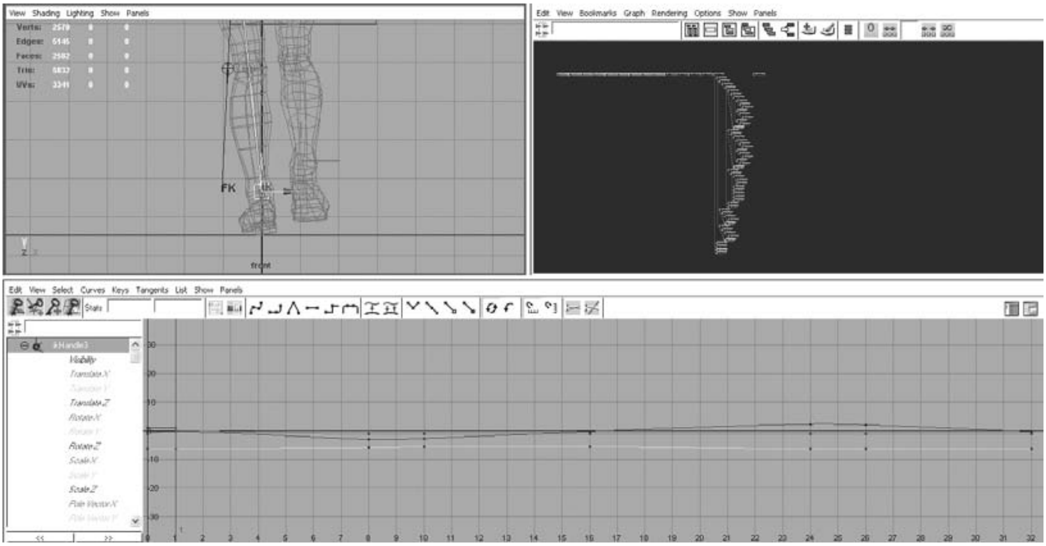


Figure 9.32
Center the right foot under the body.

no shifting of weight side to side. Because this is a game animation, however, you'll shift the feet instead of shifting the body side to side, as in video animation. To begin, go to frame 1 and change the translation of the right IK handle to 0 in the x axis, as shown in Figure 9.32.

8. At frame 16, change the translation of the left IK handle to 0 in the x axis, as shown in Figure 9.33.
9. Go to frame 32 and change the translation of the right IK handle to 0 in the x axis, as shown in Figure 9.34. Remember that frame 1 and frame 32 need to match.
10. You will notice if you play the animation back that the character's legs tend to jerk in and out. That is because there are already keyframes set at frames 8 and 10 and at frames 24 and 26. You will need to adjust these frames to get the motion to look right. First, go to frame 8 and change the translation in the x axis to .5, as shown in Figure 9.35.
11. At frame 10, change the x-axis translation to .6, as shown in Figure 9.36.
12. Adjust frames 24 and 26 in the same way so that the left foot swings in and out with out jerking.

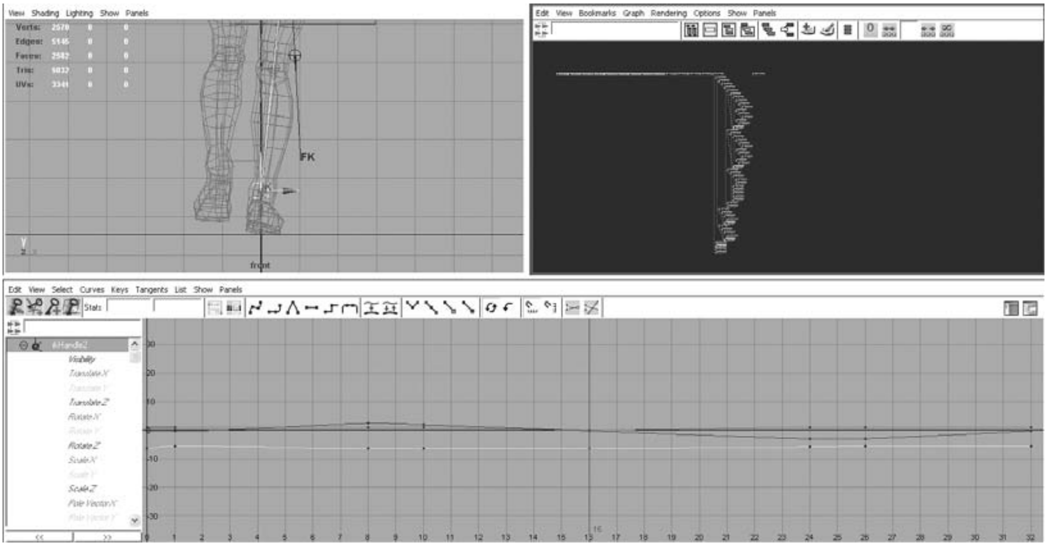


Figure 9.33
Center the left foot under the body at frame 16.

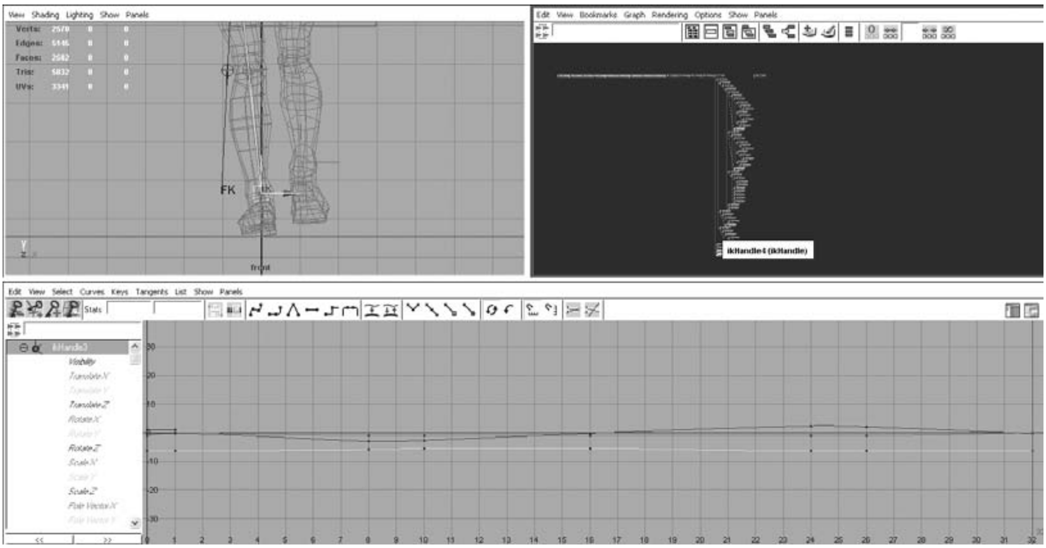


Figure 9.34
Center the right foot under the body at frame 32.

13. You will also need to take the jerking motion out of the character's right leg. Change the x-axis translation for the right foot at frames 8, 10, 24, and 26 similarly to how you changed them for the left foot except that instead of changing them .5 and .6, change them −.5 and −.6 respectively.

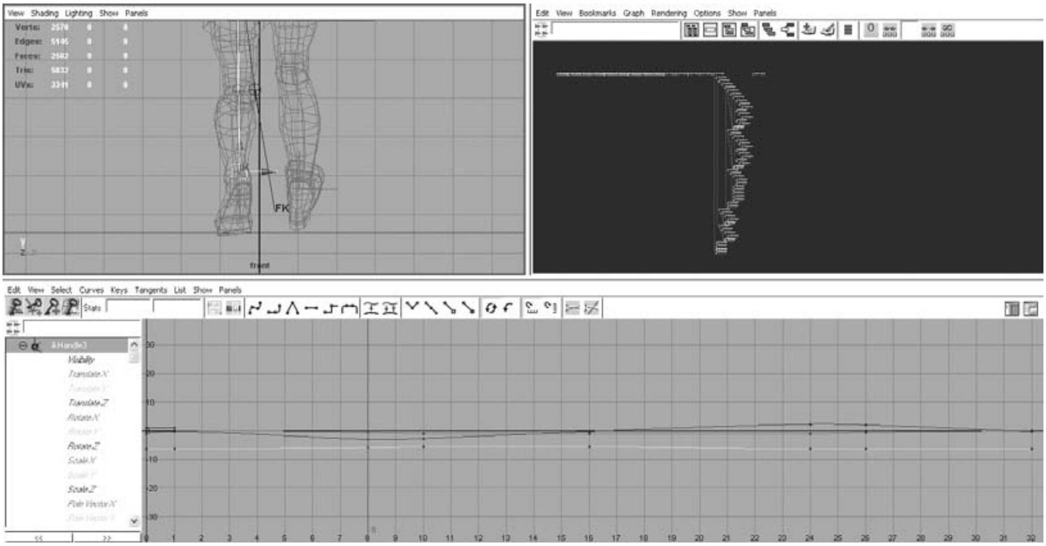


Figure 9.35
Change the left foot position at frame 8.

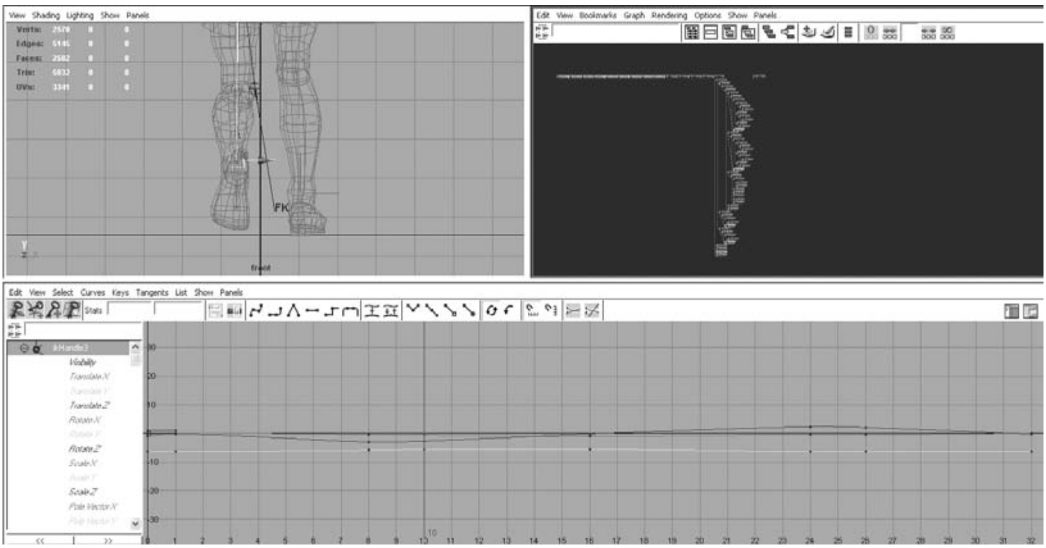


Figure 9.36
Adjust the left foot position at frame 10.

14. With all the repositioning of the feet, they are no longer well planted on the ground when they are supposed to be. To fix this, go to frame 1 and rotate the ankle joint to line up with the ground as shown in Figure 9.37.

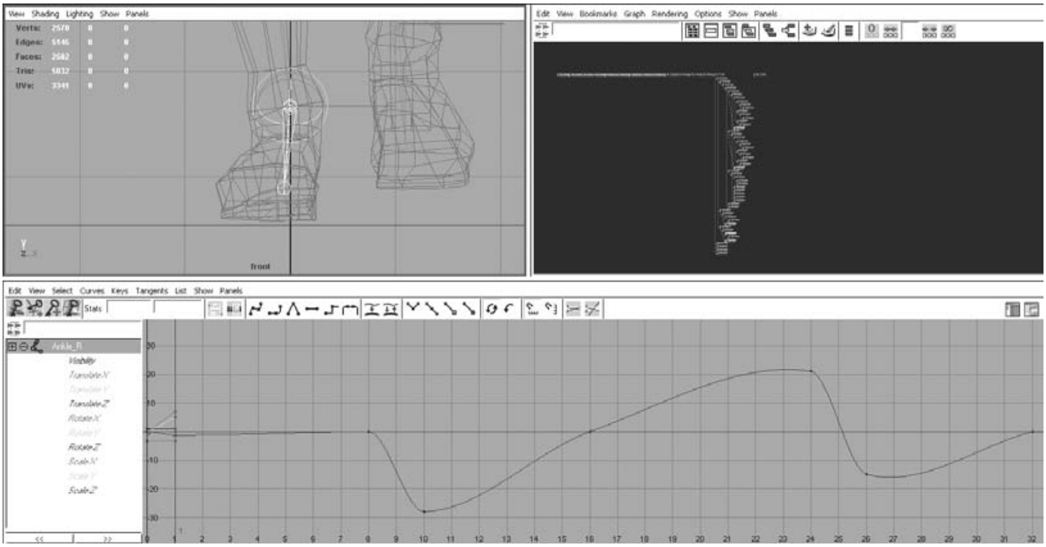


Figure 9.37
Line the foot up to the ground.

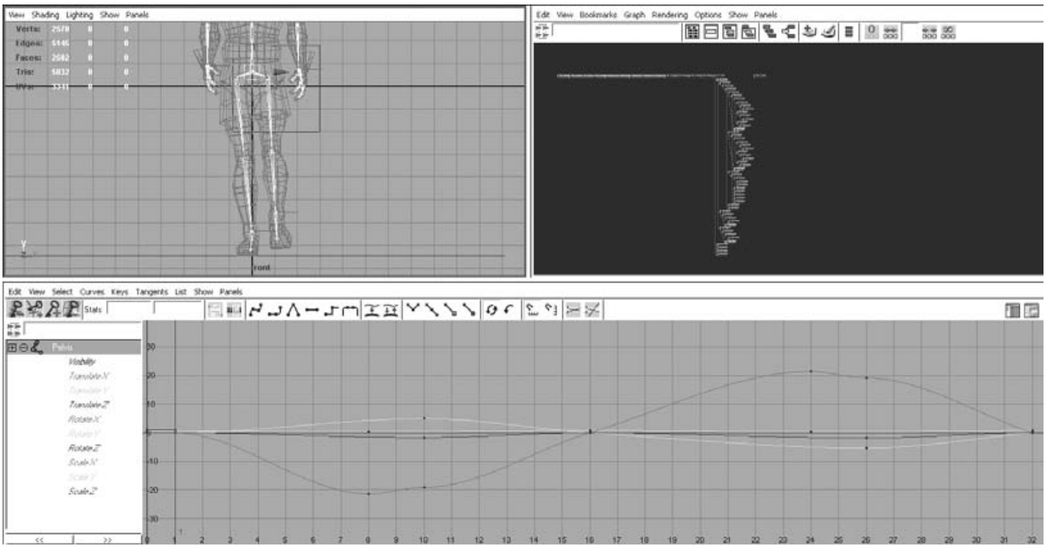


Figure 9.38
Lower the pelvis so the leg does not become hyper-extended.

15. Adjust the IK handle up or down to get the foot to rest on the ground correctly. If the leg becomes hyper-extended, you will need to lower the pelvis as shown in Figure 9.38.
16. If you lowered the pelvis on frame 1, lower it on frames 16 and 32 as well.
17. Fix the right foot so that it meets the ground correctly when it is carrying the weight of the body as shown in Figure 9.39.

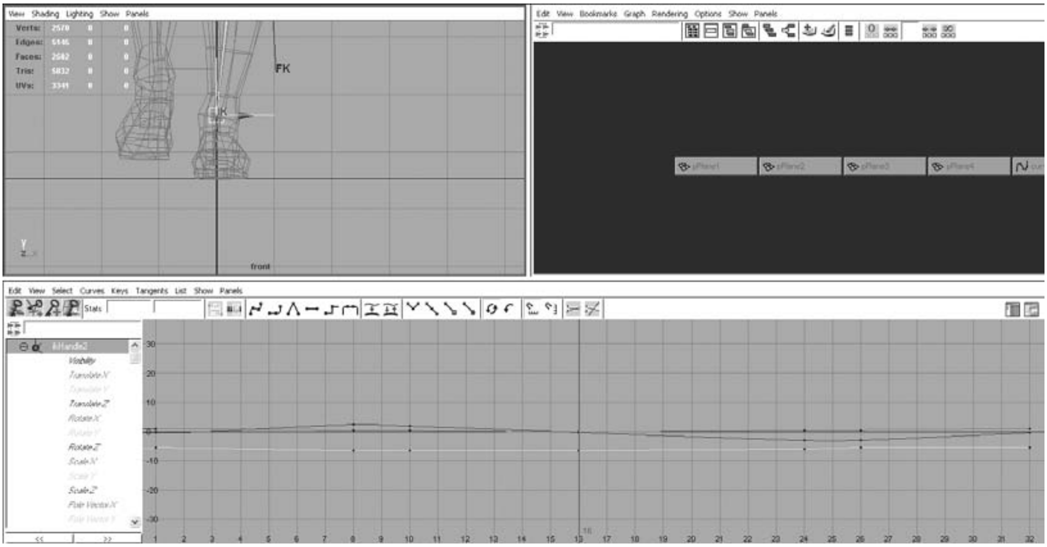


Figure 9.39
Fix problems with the right foot.

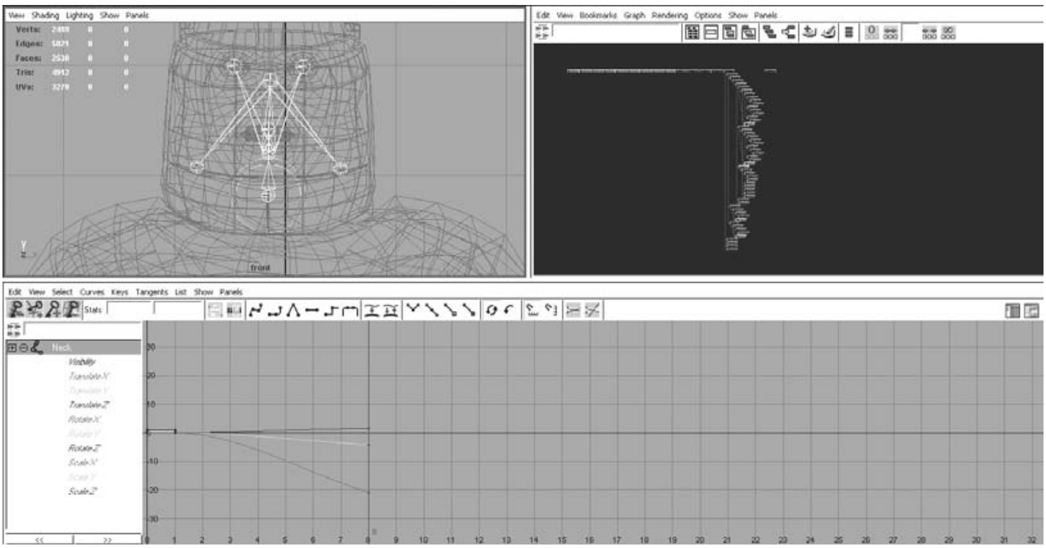


Figure 9.40
Make the character look straight ahead as he walks.

18. When a person is walking, he usually looks straight ahead. The swinging of our character's chest and hips, however, has caused him to instead look from right to left as he walks. To take this swinging motion out of the animation, first go to frame 8 and rotate the heat back so it is looking straight ahead, as shown in Figure 9.40.

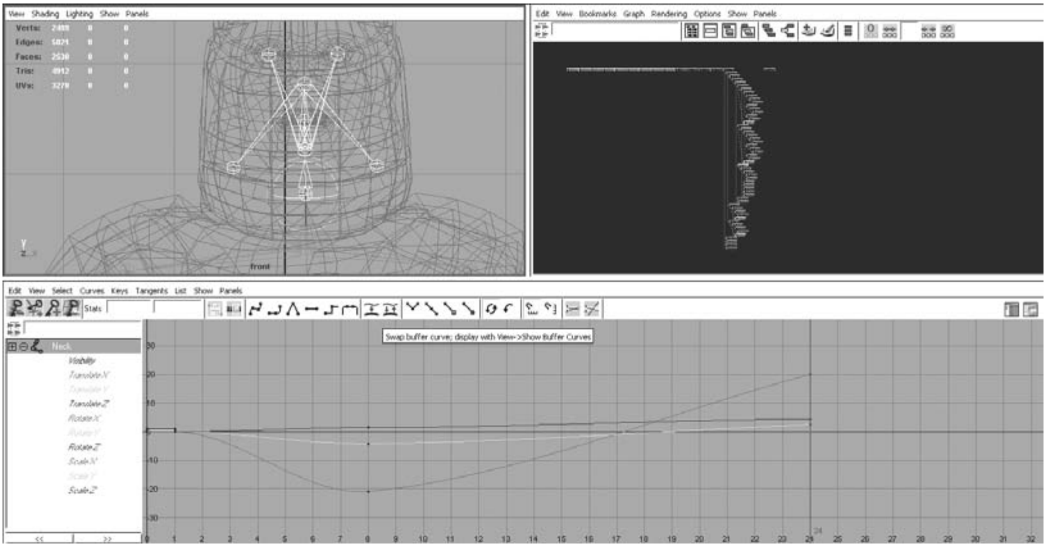


Figure 9.41
The head needs to be straightened at frame 24.

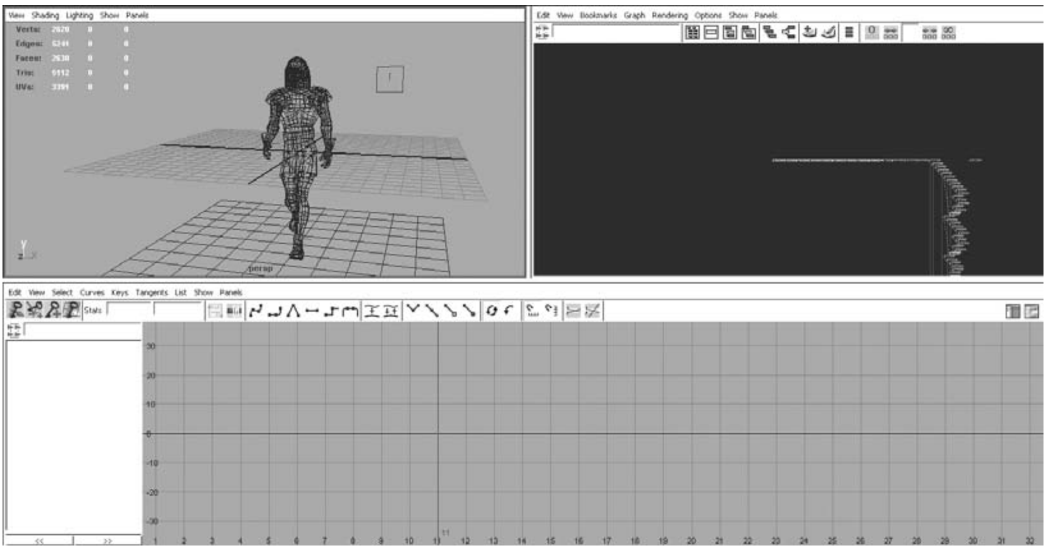


Figure 9.42
Play the animation to see if it is correct.

19. Straighten the head at frame 24 as shown in Figure 9.41.
20. Return the head to 0 rotation at frame 32 so that it matches frame 1.

The motion of the animation should look a lot better by now. Play the animation (see Figure 9.42) and see if you like it. If anything looks off, go back in and fix it.

Review those sections of the animation that pertain to any problem areas to make sure you completed them correctly.

Making Final Adjustments

The animation should be coming along well, but if you look at it in shaded mode, you will notice a few problems. For one, the arms are swinging through the shoulder plate armor as can be seen in Figure 9.43.

To fix the problem, you need to rotate the shoulder plate back as shown in Figure 9.44.

There is also a problem with the arms jerking at frame 2. This is because of the tangents coming from the bind pose at frame 0. To fix this problem, break the tangent on frame 1 for the right and left IK handles on the arms as shown in Figure 9.45.

One other problem I noticed with my animation was that the character tended to look like he was leaning backward as he walked. To fix this problem, I first adjusted the head as shown in Figure 9.46.

After working on the head, I noticed that there were still problems, so I also adjusted the torso as shown in Figure 9.47.

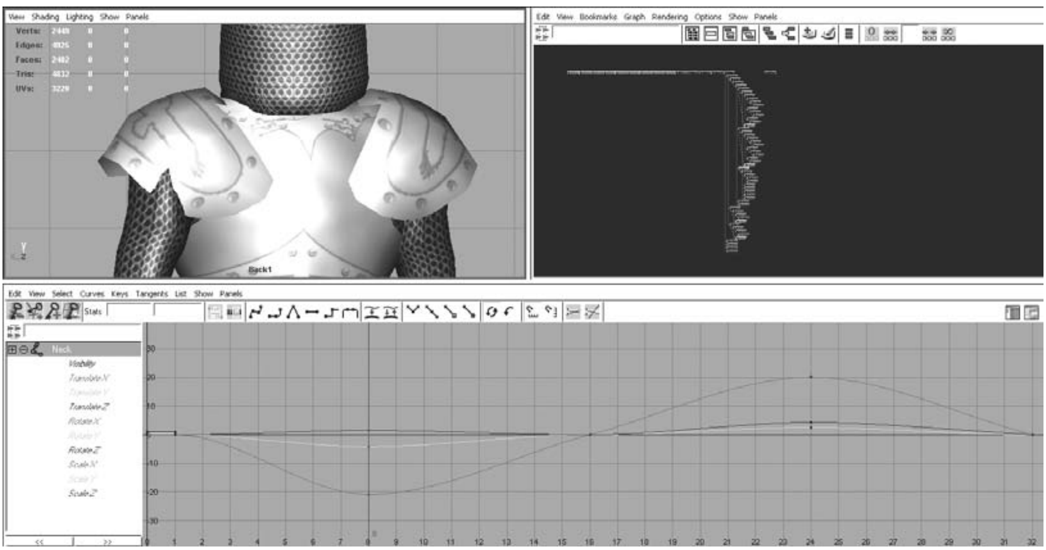


Figure 9.43
The arm swings through the shoulder-plate armor.

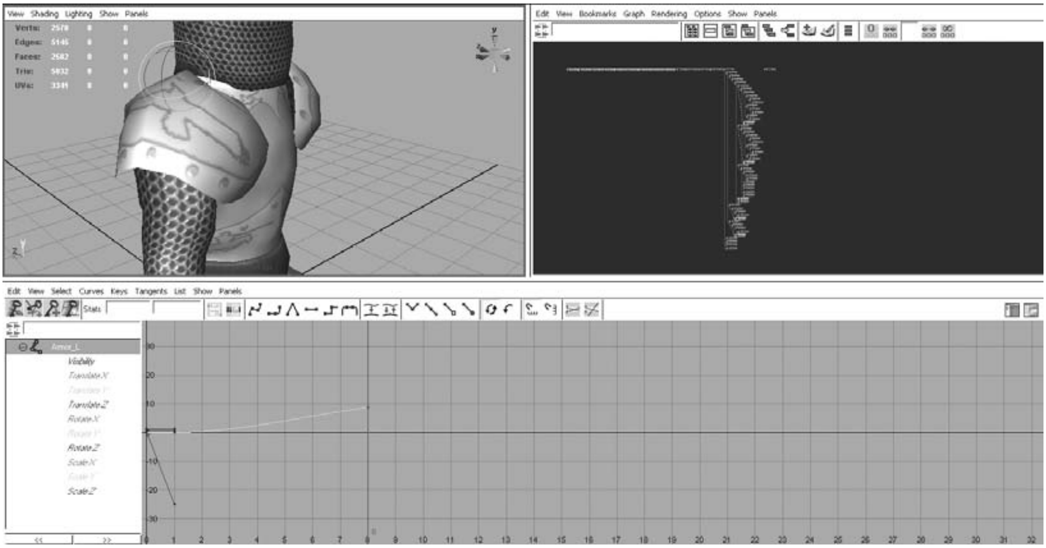


Figure 9.44
Rotate the shoulder plate armor back.

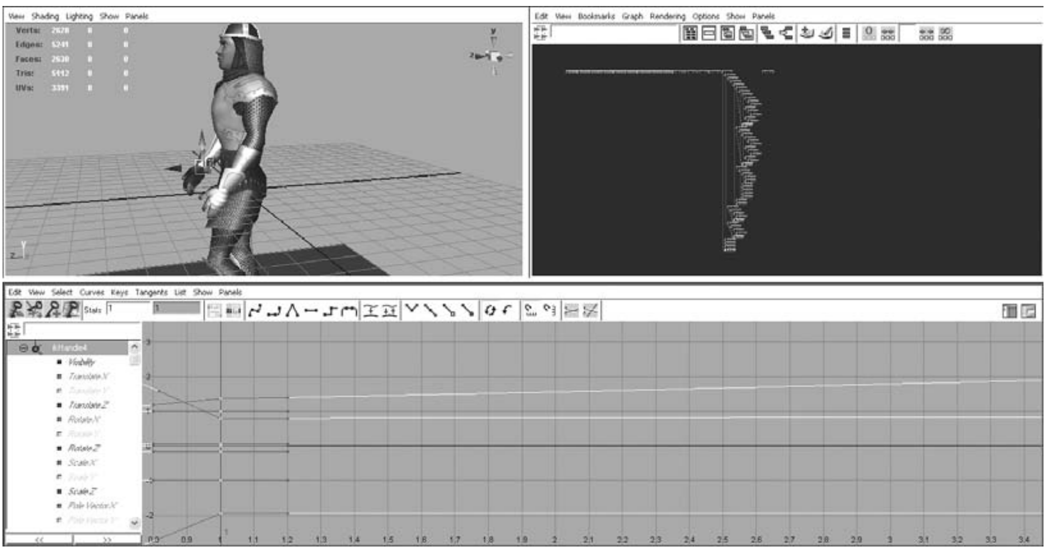


Figure 9.45
Break the tangent on the IK handles at frame 1.

There may yet be some minor adjustments that could be made to make the animation look more real, but what you have now is a basic walk animation. I hope this project has helped you understand how to animate cycles in Maya and how to get a good walking motion for your characters.

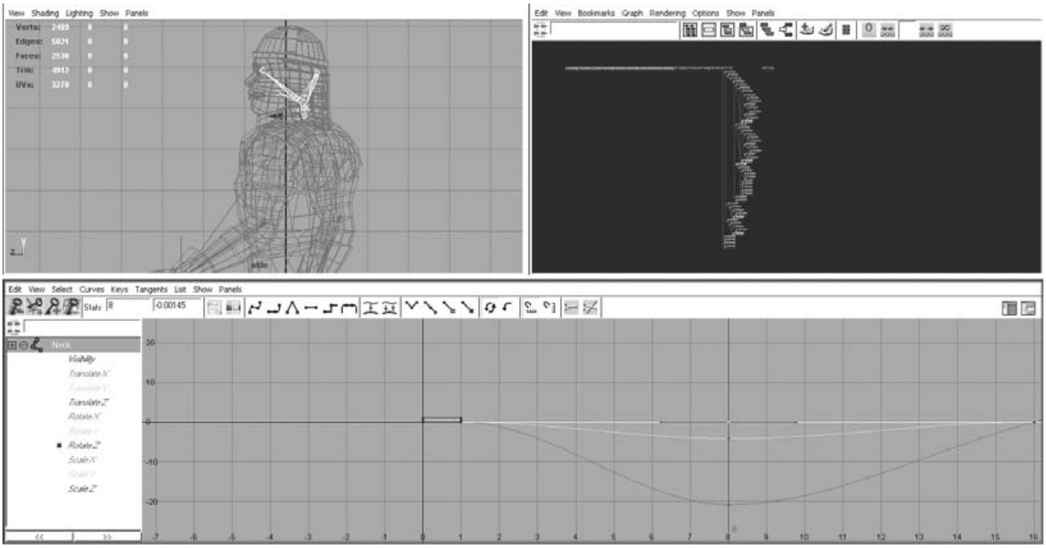


Figure 9.46
Rotate the head forward where it tends to jerk back.

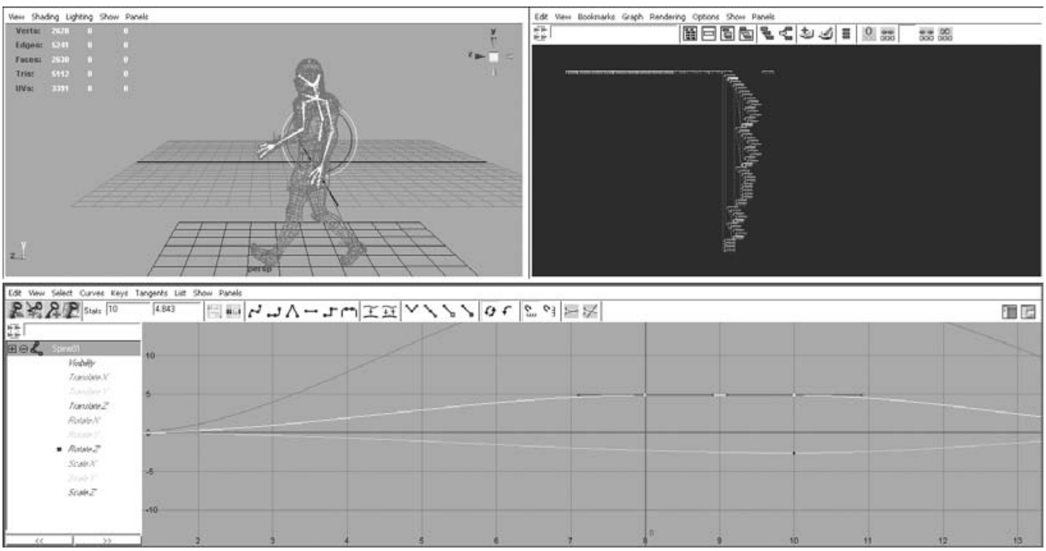


Figure 9.47
Adjust the torso so the character leans forward during the extended strides of his walk.

Note

A walk animation is a complicated animation and requires a great deal of work to get it to look right. If you would like to have the animation file with the character shown in this chapter, it is available on http://www.turbosquid.com. (Look it up under "Pardew Studios.") Studying the file may help you if you have problems following the instructions in this chapter.

Summary

In this chapter you learned how to make a walking cycle animation for games. Animating the walking cycle for game characters is different from animating for video or film in that the character moves as though he is walking in place.

In this chapter you learned the following:

- Animating in place
- Moving the legs
- Animating the chest
- Animating the arms
- Animating the feet
- Animating in 3D
- Animating details
- Fixing animation problems

CHAPTER 10

Non-Linear Animation

Non-linear animation refers to moving, blending, and working with animation clips interactively rather than just putting them end-to-end. In non-linear animation, clips of animation are created, which are sequences for key frames taken from the animation sequences you create for your characters. Maya has a great tool for non-linear animation called the *Trax editor*. Using the Trax editor, these clips can be moved, modified, and blended without destroying the original animation.

The power of non-linear animation is that it allows the animator to create animation sequences and store them one at a time. These animation sequences can then be blended and combined for different new animations. For example, a walking animation can be blended with a looking animation so that as a character walks, he can look around without the animator having to create a new walk/look animation. The finished set of animations can be exported individually or as a set, and the frame data for beginning and ending each sequence is automatically stored in the file. To illustrate how non-linear animation works, this chapter examines the Trax editor, and then demonstrates how to use it by stepping you through the creation of some animation clips based on the walk animation you created in the last chapter.

Note

Although not all game engines support non-linear animation, enough do to merit a discussion of it here.

Using the Trax Editor

The Trax editor allows for great flexibility in animating characters. It treats a character as a single unit called a *character set* and stores animation sequences as *poses* or *clips*. Using the Trax editor, you can mix and match animation sequences in any order you want and blend motion between them. This section shows you how to use the Trax editor.

To launch the Trax editor, shown in Figure 10.1, open the Windows menu, choose Animation Editors, and select /Trax Editor. Notice that the Trax editor is similar to the Graph editor in that time is represented from left to right. It is different from the Graph editor, however, in that it vertically lists characters and clips instead of motion paths.

Before you can use the Trax editor, you need to first load a character set into the editor. *Character sets* in the Trax editor are groups of objects that can be animated. These objects can be all the joints in a character or they can be only some of the joints. Character sets can also have sub-character sets, where a group of joints such as the character's hips and legs are a subset of the character's main set.

1. Select all the nodes in the bone tree of your warrior character. The bone tree will be used to create the warrior's character set.
2. In the Trax editor, open the Create menu and choose Character Set to open the Create Character Set Options dialog box, shown in Figure 10.2.
3. Type `Warrior` in the Name field to name the character set.

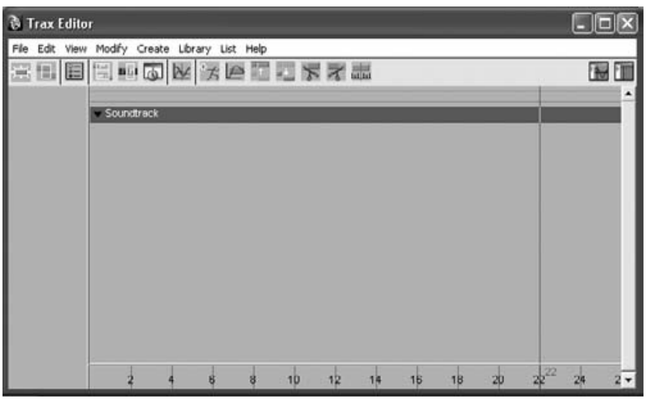


Figure 10.1
The Trax editor is similar to the Graph editor.

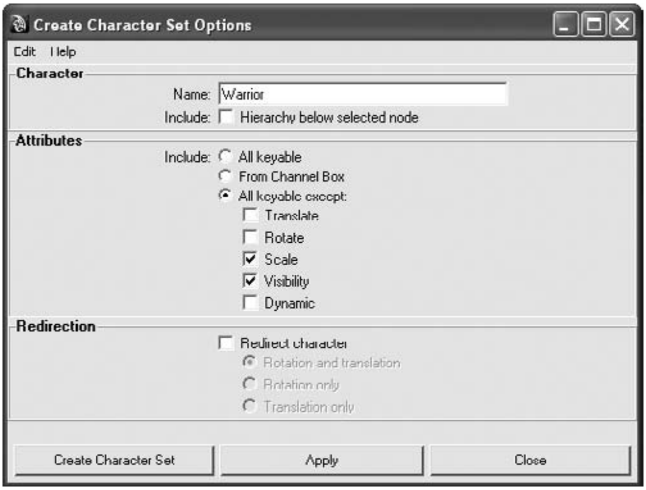


Figure 10.2
Create a character set within the Trax editor.

4. Leave the default settings as is and click Create Character Set.

Note

The default settings for character sets in the Trax editor include Translate, Rotate, and Dynamics in the grouping, while excluding Scale and Visibility. That's because most animation does not deal much with scale or visibility; excluding them improves performance while working on the animation. Scale and visibility are advanced animation properties and they can cause problems for beginning animators. If you are an advanced animator and know how to apply scale and visibility to your animations, you can select those attributes as well.

5. Load the Warrior character set into the Trax editor. To do so, open the drop-down list to the left of the Auto Key button and choose Warrior. Next, click the Load Selected Characters button to load the Warrior character set into the Trax editor (see Figure 10.3). You are now ready to use the Trax editor to animate your character.

Creating Clips

To create a clip, do the following:

1. Move the walk animation from the timeline to the Trax editor. To do this, open the Create menu in the Trax editor and choose Animation Clip to open the Create Clip Options dialog box shown in Figure 10.4.

2. To name the clip, type `walk` in the Name field.

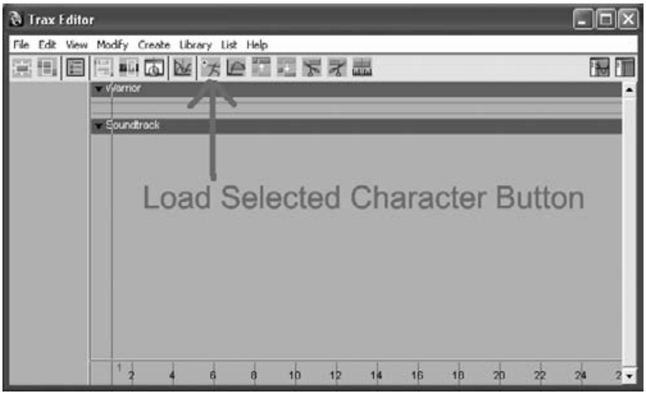


Figure 10.3
Load your character set into the Trax editor.

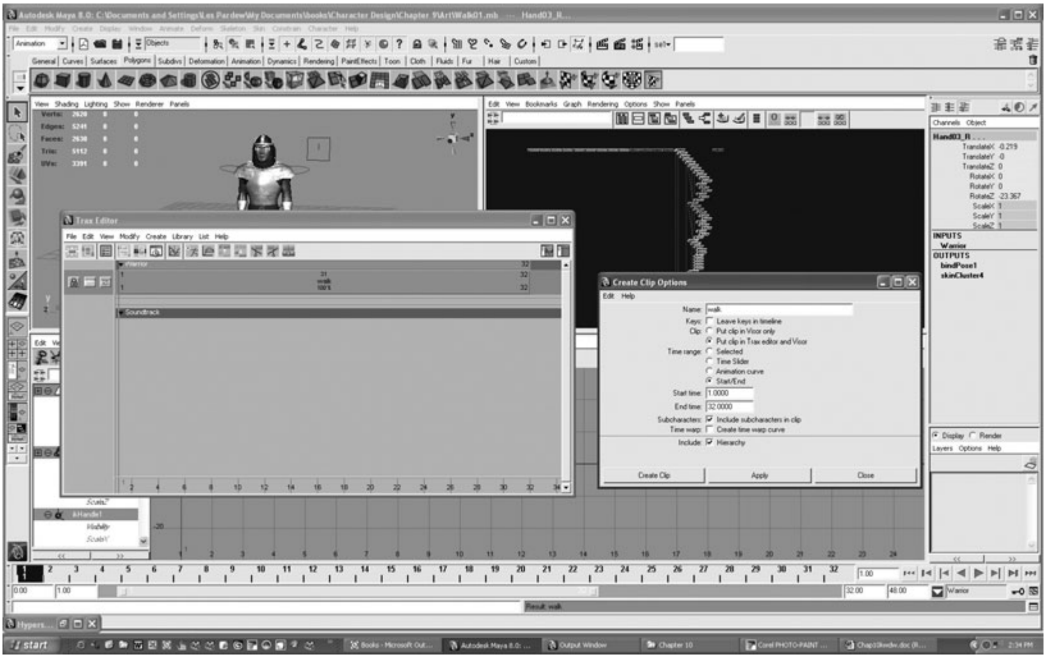


Figure 10.4
Put the walk animation clip in the Trax editor.

3. In the Start Time field, type 1; type 32 in the End Time field. This indicates that the clip should run from frame 1 to frame 32.

4. Click Apply to move the animation clip from the timeline to the Trax editor.

Note

Actually, Maya makes two copies of the animation: a source clip and an instant. Only the instant is moved to the Trax editor. The source clip is stored so that it remains untouched by any modifications to the animation in the editor.

5. Slide the Current Frame indicator along the timeline; notice that the character still walks just like before. That's because the clip is enabled.
6. Right-click the clip and choose Enable to toggle it off. This will allow you to create a second animation sequence.

Note

It is a good rule to not have any animations enabled when you start working on a new animation. Otherwise, the enabled keys will mix with the new ones, creating unworkable conditions for animating.

Ready Animation

To really understand how to use the Trax editor, you need to have more than one animation clip. Let's create a second animation clip so you can see how to use them in the Trax editor together. In this second animation, the character will begin in the starting walking position and then assume a ready position, as if expecting a fight. First, minimize the Trax editor and set a key frame for all of the nodes in the bone tree at frame 1, as shown in Figure 10.5. Then do the following:

Note

It is always a good idea to set a key frame for all the joints and IK handles at the beginning of the animation even if you will be modifying them later. By setting a key at the beginning of the animation, you lock all of the animation attributes so they don't change unless you change them.

1. Move the Current Frame indicator to 1.
2. Select the character's pelvis and move it down just a little so the character's knees bend.
3. Rotate the pelvis to the right so the character's left hip and shoulder are forward, as shown in Figure 10.6.
4. Move the character's right leg back and his left leg forward, and adjust the feet to line up with the floor as shown in Figure 10.7.

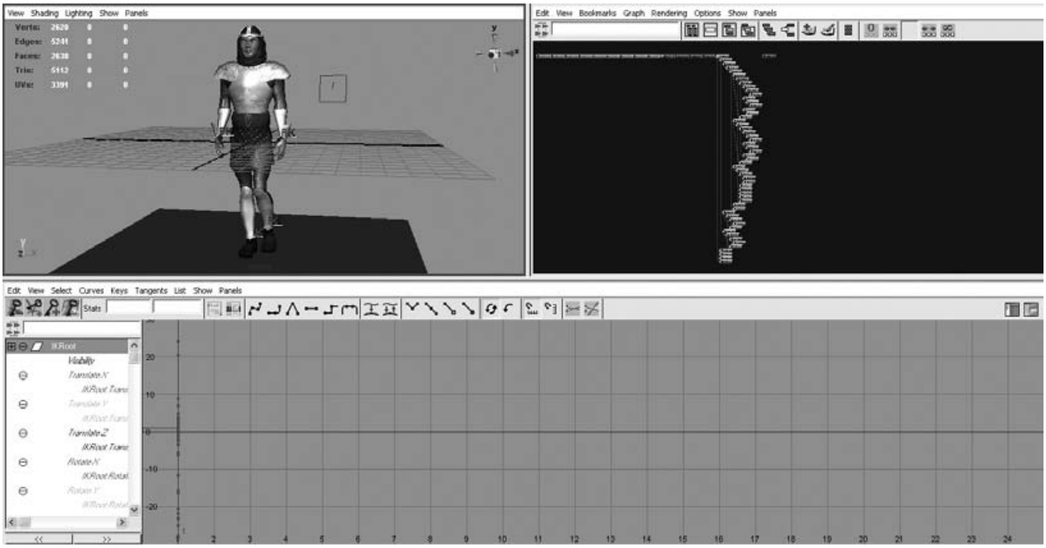


Figure 10.5
Set a beginning key frame.

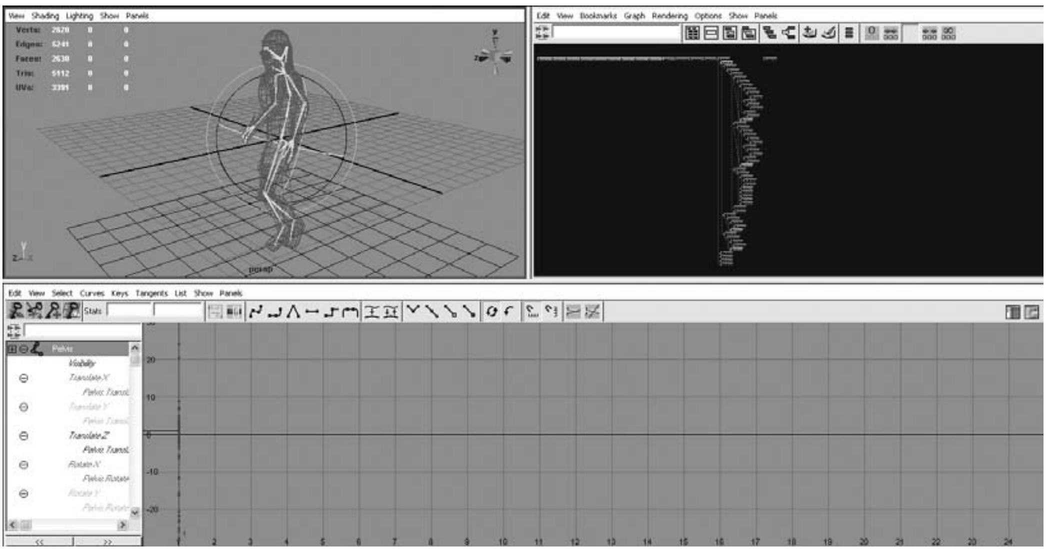


Figure 10.6
Rotate and lower the pelvis.

5. In the ready position, the character will have his hands in fists. Rotate the finger joints of the character so his hands are fisted, as shown in Figure 10.8.

6. Next, raise the character's arms up as if to block incoming punches. As you move the characters arms up into a ready position, you may find that the

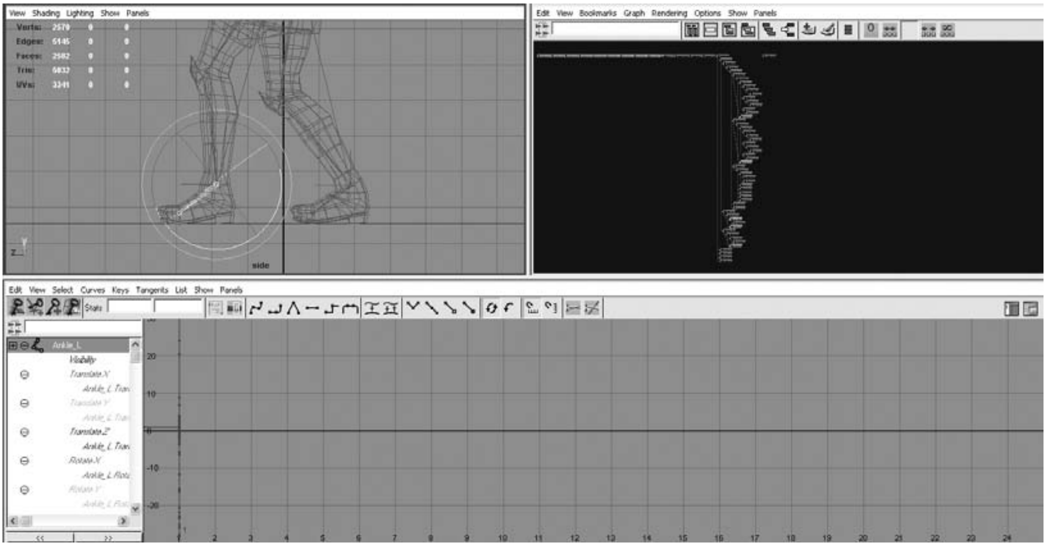


Figure 10.7
Position the feet on the ground.

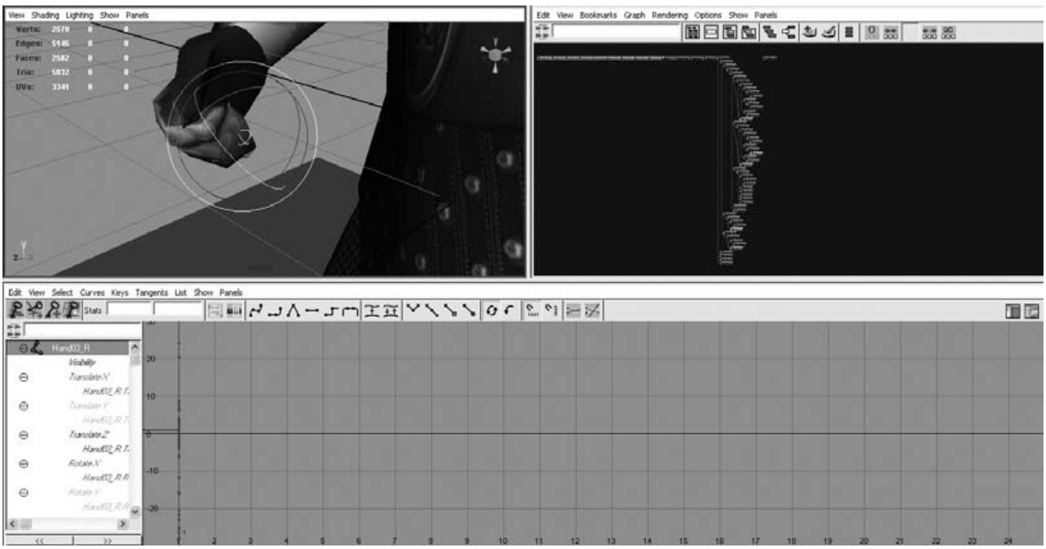


Figure 10.8
Curl the character's hands into fists.

angle of the elbow is not right. You don't want to rotate the shoulder joint because it has an IK handle attached to it. Instead, select the IK handle and then select the Show Manipulator tool from the Toolbox. A circle will appear around the IK handle; it controls the pole vector of the IK chain. You can use it to rotate the elbow to a position you like, as shown in Figure 10.9.

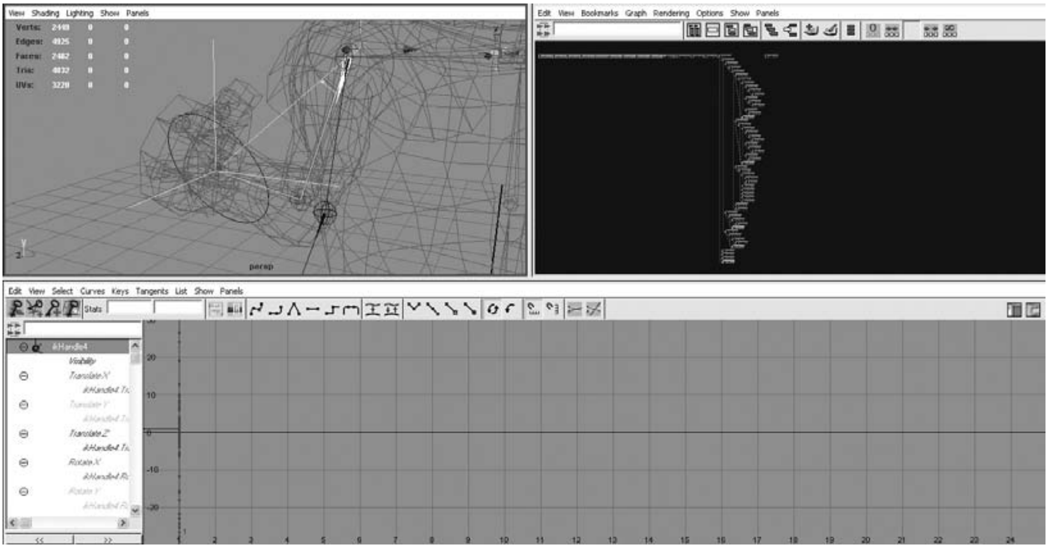


Figure 10.9
Rotate the arm by rotating the pole vector.

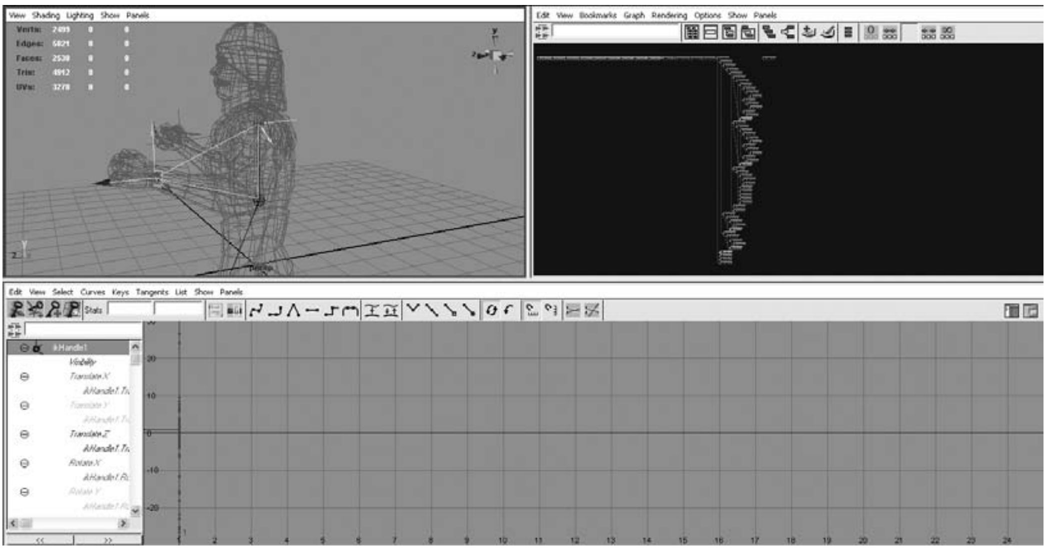


Figure 10.10
Use the IK handles and the Move tool to position the arms.

7. Move the arms up into a ready position, as shown in Figure 10.10.
8. The character looks a little off balance. To rectify this, rotate the spine from the front view to be more vertical, as shown in Figure 10.11.

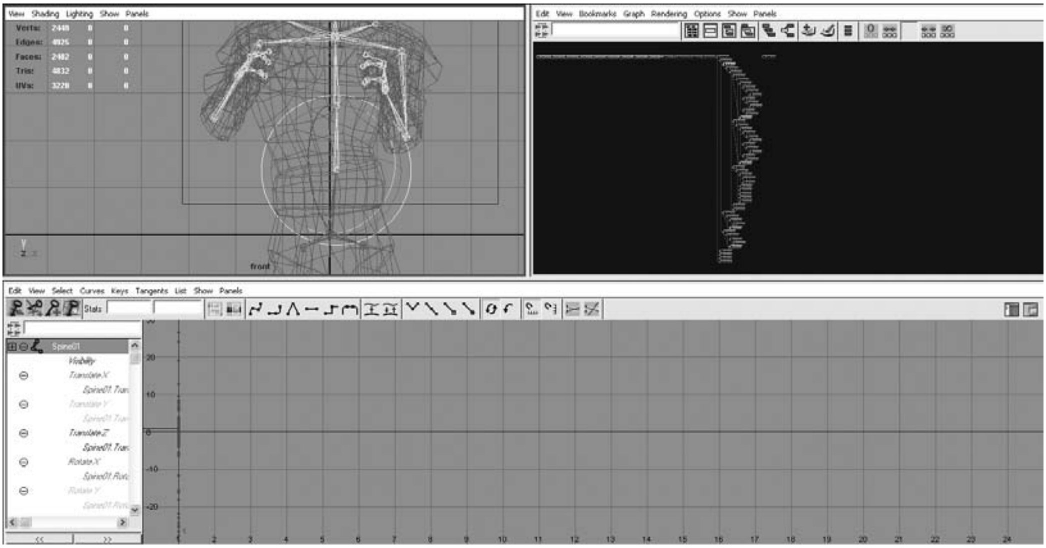


Figure 10.11
Rotate the spine to be more vertical.

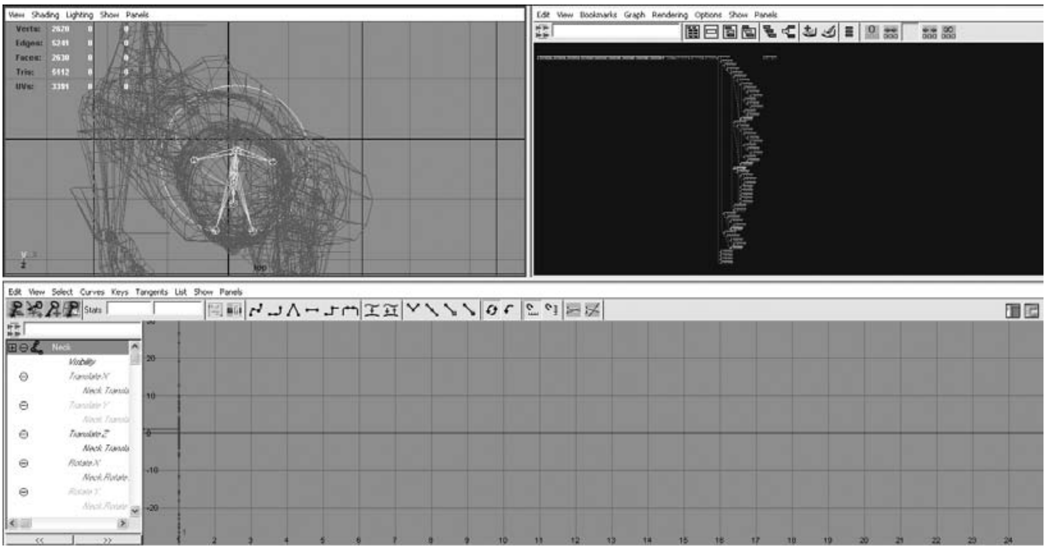


Figure 10.12
Rotate the character's head forward.

9. From the top view, rotate the character's head so he is looking straight ahead, as shown in Figure 10.12.

10. Rotate the character's pelvis forward so he leans forward a little, as shown in Figure 10.13.

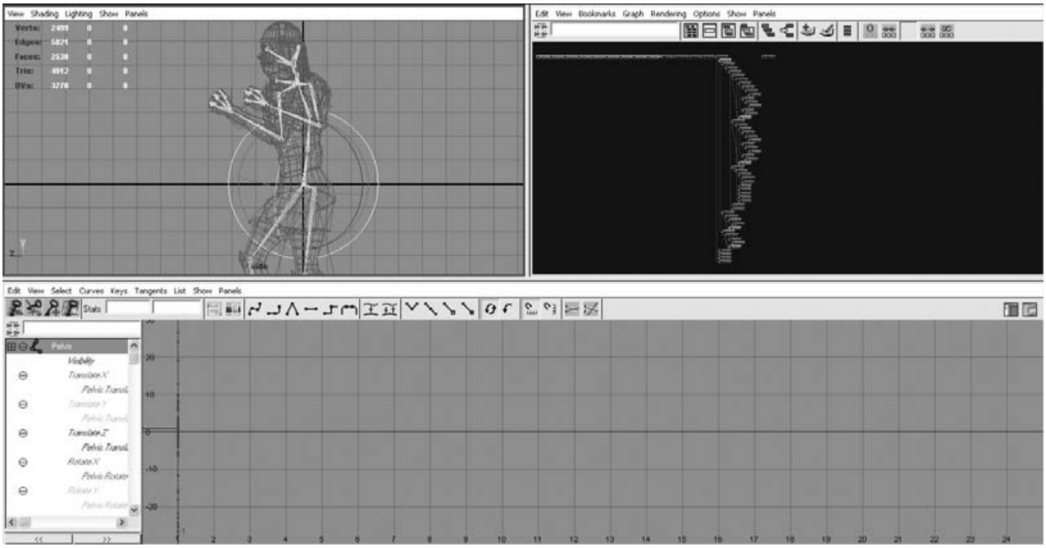


Figure 10.13
Rotate the character's body forward.

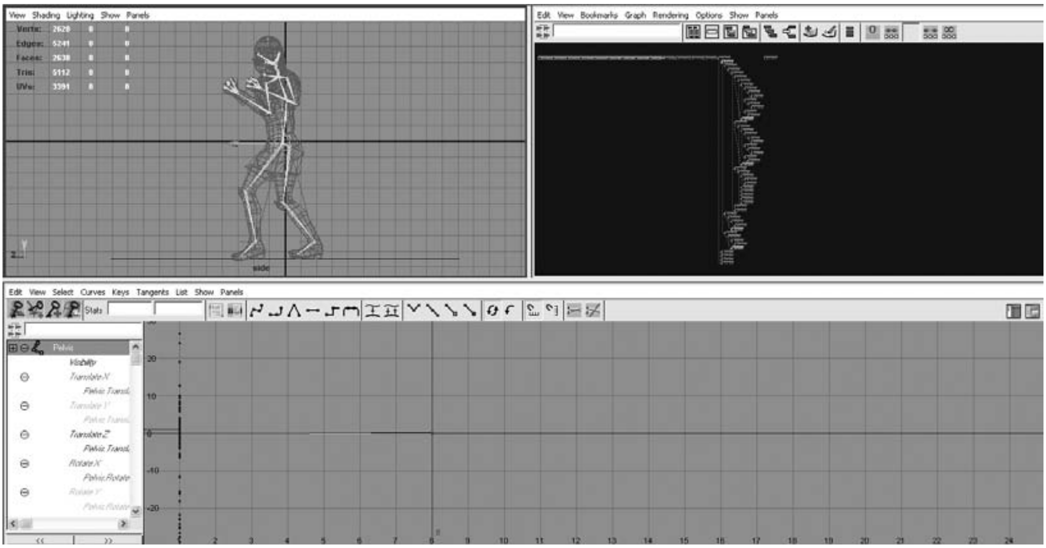


Figure 10.14
Move the pelvis forward and down.

11. Next, put a little bounce in the character so he looks like he is alive and ready to fight. To achieve this, you'll move the pelvis in a circular motion. To begin, switch to the side view and select the pelvis.

12. At frame 8, move the pelvis down and forward just a little, as shown in Figure 10.14.

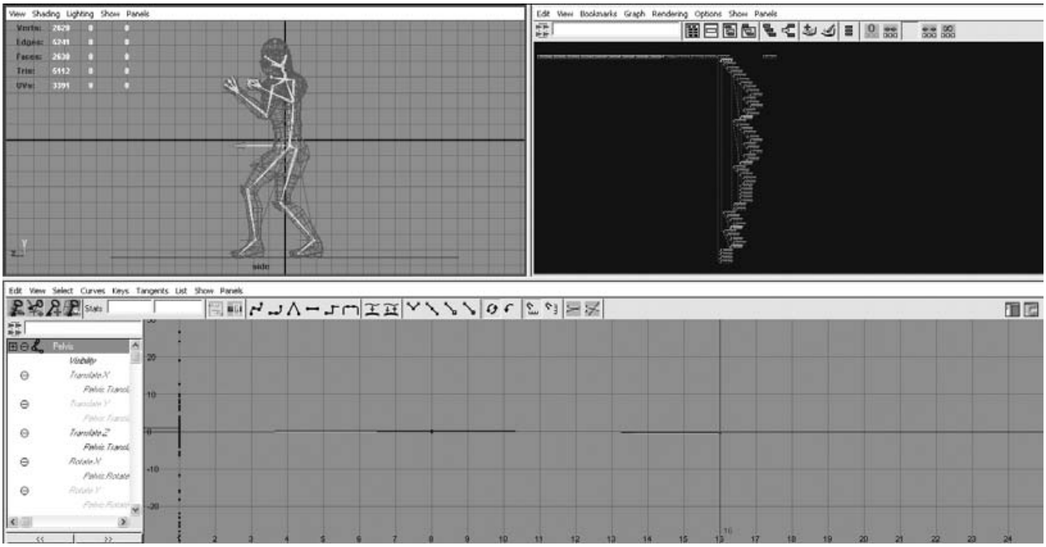


Figure 10.15
Swing the pelvis down and back at frame 16.

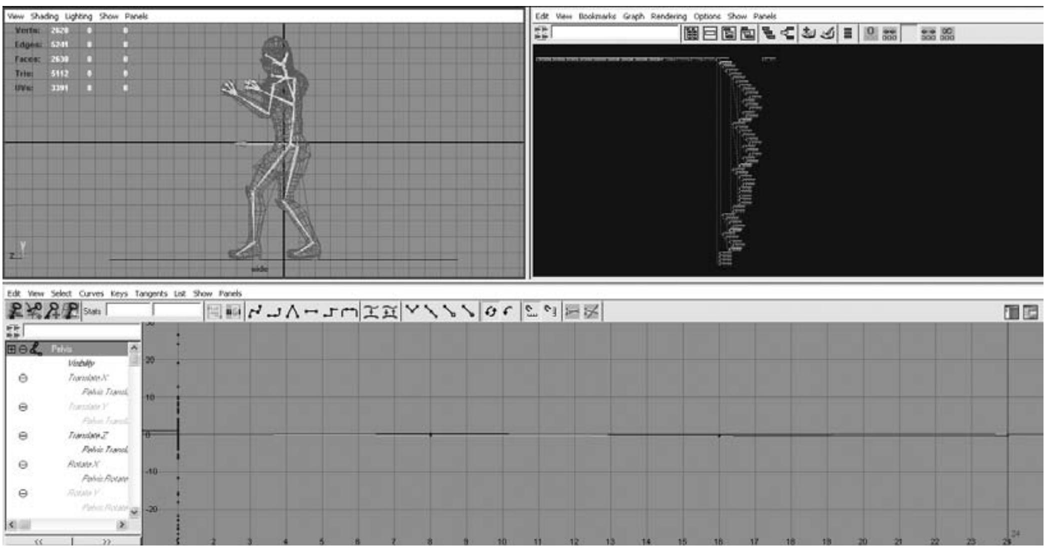


Figure 10.16
Move the pelvis up and back a little more at frame 24.

13. At frame 16, move the pelvis down a little more and pull it back, as shown in Figure 10.15.
14. At frame 24, pull the pelvis back a little more and up a little, as shown in Figure 10.16.

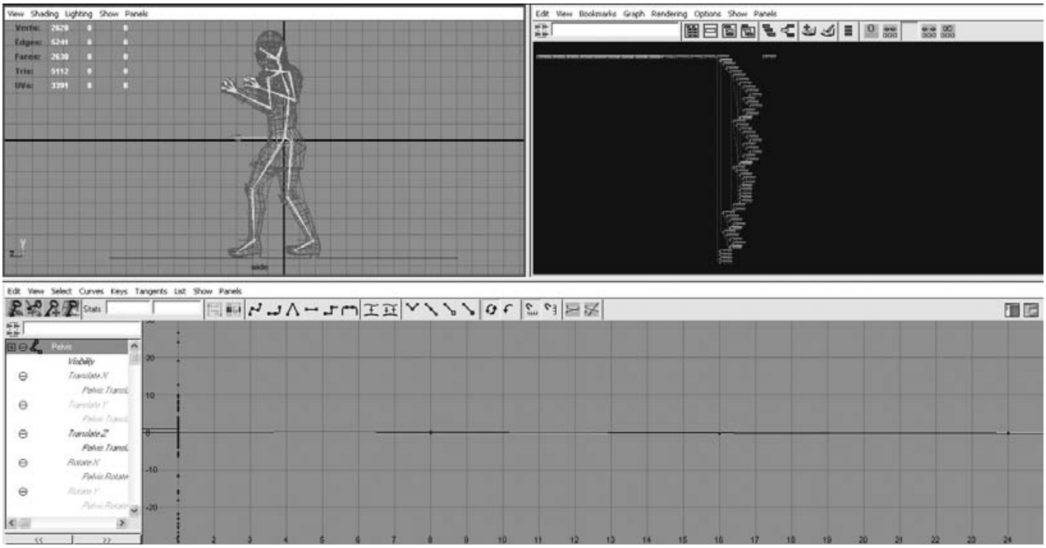


Figure 10.17
Match frame 32 with frame 1.

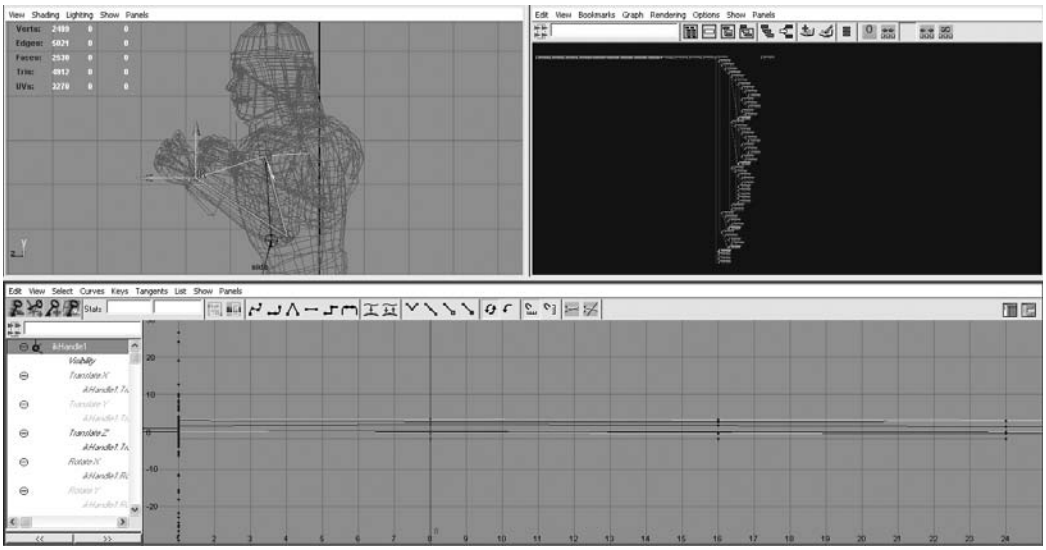


Figure 10.18
Have the fists move in a circular motion.

15. At frame 32, match the position of frame 1 so the motion will cycle. Use the Channel box to match the translations for both frames, as shown in Figure 10.17.
16. Use the IK handles to move the character's fists in a circular motion, as shown in Figure 10.18. This motion should be like the pelvis's, except the

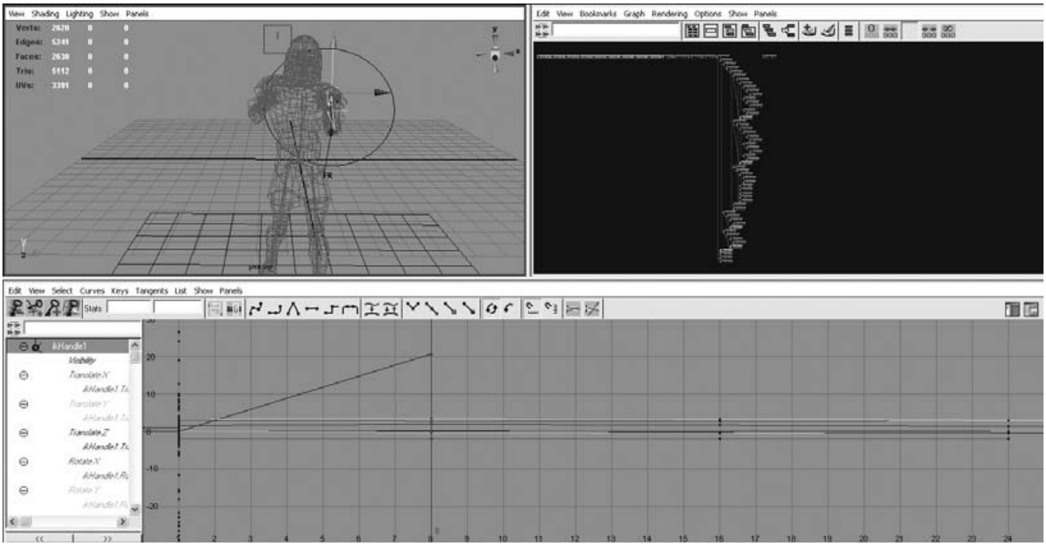


Figure 10.19
Make sure the elbows don't sink into the torso.

fists need to be offset from the pelvis and from each other to look natural. Have one fist come up when the pelvis moves down and the other move back when the pelvis moves forward.

17. You may need to adjust the pole vectors of the arms as shown in Figure 10.19 so they don't sink into the torso.

18. You need just a little motion in the shoulders. Rotate the left shoulder a few degrees forward at frame 8 and back at frame 24, as shown in Figure 10.20.

19. Go back to the side view and adjust the feet so they stay planted on the ground, as shown in Figure 10.21.

20. Adjust the animation where needed in the Graph editor and then move the animation sequence to the Trax editor as shown in Figure 10.22. Label the clip Ready.

21. Now you have two animation clips in the Trax editor. Move the Ready clip to the end of the Walk clip and enable both clips to play the animation (see Figure 10.23). When you're finished, save the file.

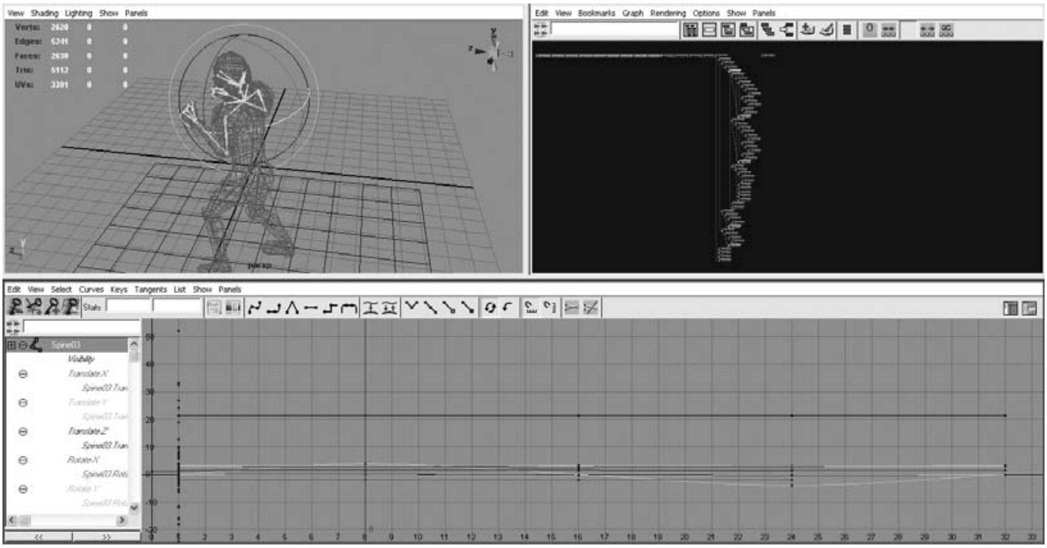


Figure 10.20
Add a little movement in the shoulders.

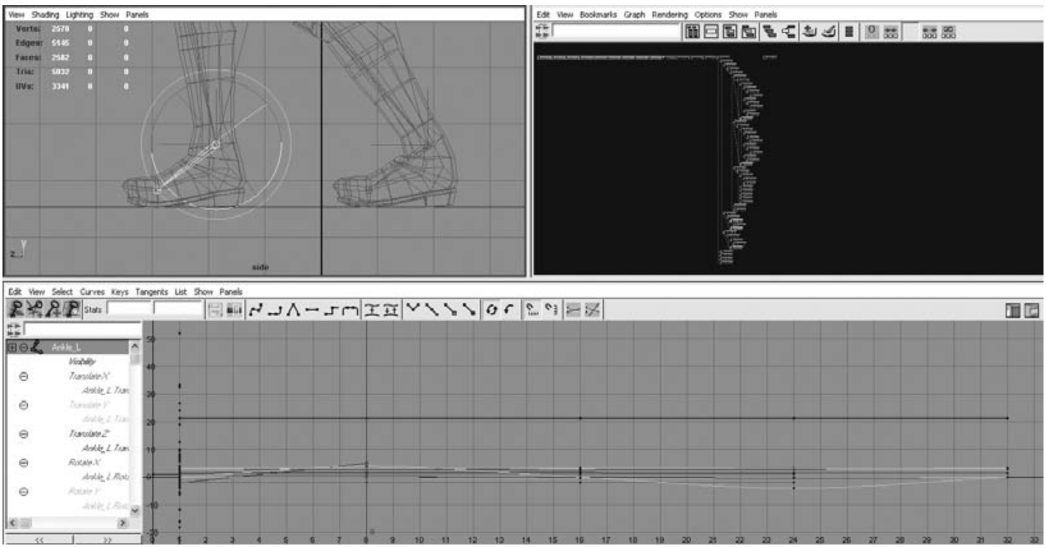


Figure 10.21
Adjust the feet.

The Trax editor will allow you to move an animation to any time within a timeline. Be careful about having two animations share the same time slots, however. When this happens, the animation data is combined and weird things will start to happen to your animations. You can avoid problems by turning

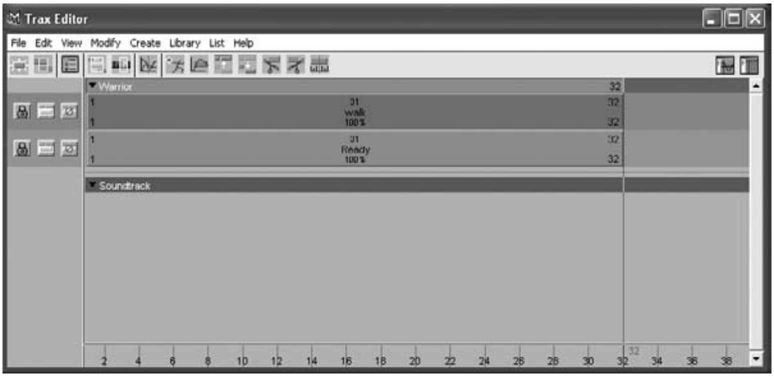


Figure 10.22
Move the finished animation sequence to the Trax editor.

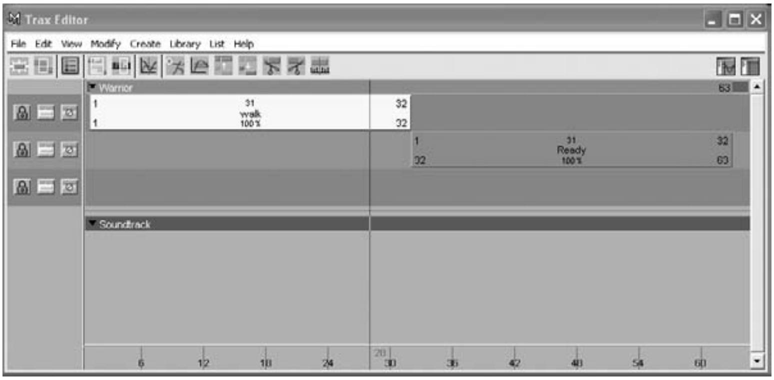


Figure 10.23
Play the two animations end to end.

Enable off in the marking menu. To do so, move the cursor over a clip and hold the right mouse button down; the marking menu will appear. If there is an × in the Enable checkbox, the clip is enabled. Click the Enable checkbox to deselect it. (I always move my clips around with Enable off and then turn it on for playback only.)

If you store a clip in the Trax editor and later decide you want to change the animation, you can right-click the clip and activate the keys. This puts the keys back in the timeline; you can manipulate them just as if they were not in the editor. When you are finished, you don't need to create a new clip in the Trax editor for the changed animation; just toggle off Activate Keys, and the animation clip is ready to go.

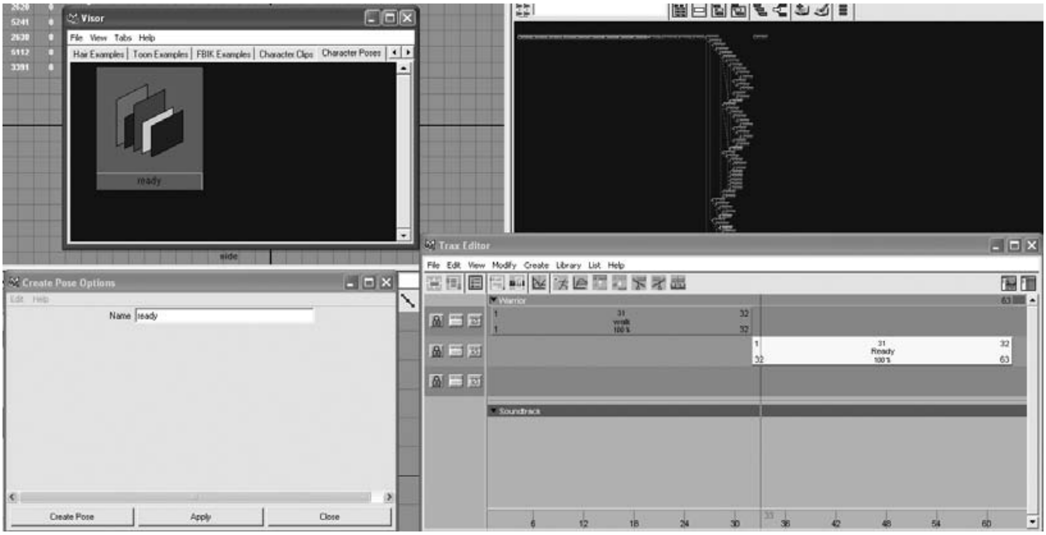


Figure 10.24
Create a pose from the Ready animation.

Blending Clips

Another nice feature of the Trax editor is its ability to blend the animation of two clips together. Let's take a look at how the blending feature works, as well as how you can use poses to create animations in the Trax editor. Go back to your warrior project and open the Trax editor with the Warrior character loaded.

1. Go to the first frame of the Ready animation and create a pose from that frame as shown in Figure 10.24. Poses are stored in the visor, also shown in the Figure. You can store as many poses as you like and use them to help when creating animations. Name your pose Ready.

2. Drag your two animation clips to the right so they start on frame 10 (see Figure 10.25).

3. Move the Current Frame indicator to frame 1.

4. Right-click the pose in the Visor and select Apply Pose from the menu to apply the pose to the character at frame 1 (see Figure 10.26).

5. Disable both clips to clear the animation timeline. To begin, create a new animation clip that is two frames long. Name it Ready1, as shown in Figure 10.27.

6. Select Ready1 and, while pressing the Shift key, also select walk.

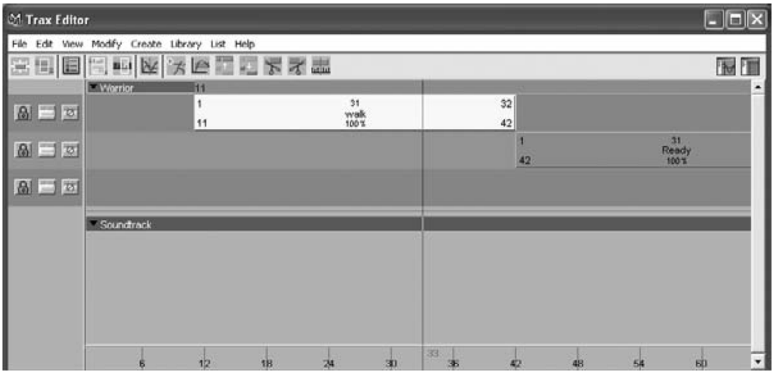


Figure 10.25
Move the two animation clips to the right.

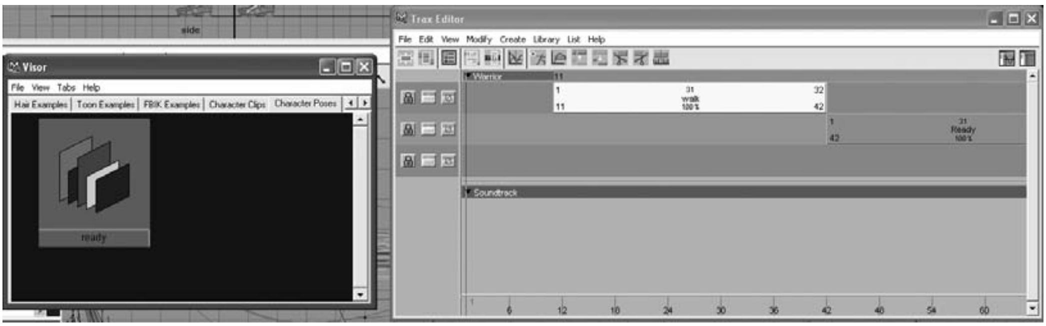


Figure 10.26
Apply the Ready pose at frame 1.

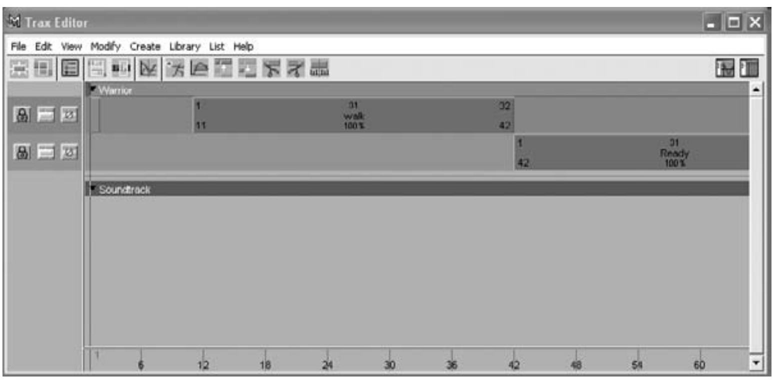


Figure 10.27
Create a new animation clip and call it Ready1.

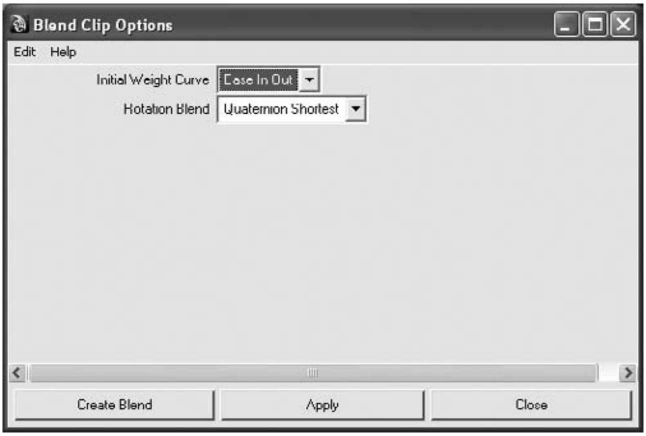


Figure 10.28
Blend the two animation clips.

7. Open the Create menu and choose Blend to open the Blend Clip Options dialog box.
8. Change the Initial Weight Curve setting to Ease In Out and click Apply to blend the two clips, as shown in Figure 10.28.
9. Enable the walk clip and play the animation to see how the blending works. Remember to enable the clips.

The two clips blend into each other to create a transition animation between the ready pose and the walk animation. You can also blend the walk animation with the original ready clip on the other end so it blends with that animation as well. Enable the Ready clip and then blend the two clips.

You will notice that when you play the animation between the two clips, there doesn't seem to be any blending. That is because the clips are right next to each other in the frame sequence. Move the Ready animation a couple of frames to the right as shown in Figure 10.29 and you will start to see the blend.

This points out another great feature of the Trax editor: The blends are dynamic and will expand over as many frames as you want them to.

There are a number of other features in the Trax editor for working with clips. You can trim, split, and merge clips. You can group and ungroup clips. You can even copy, paste, and duplicate clips. With a little practice you will be working in the Trax editor with no problem at all.

Figure 10.29
Expand the blend between the two clips.

Exporting Animation

So far, this book has covered many aspects of developing game art, but the book would not be complete if it didn't talk a little about exporting data to a game engine. Seldom will a game engine accept native Maya files. Most of the time, the art must be converted into a specific format for the game. Game engine formats are generally streamlined and contain only essential data that the engine actually uses. Native Maya files, on the other hand, contain extensive data, some of which is not needed in the engine.

Because game engine formats contain limited data, some features in Maya do not export to the game. It is vital that game artist study the features that do export and those that don't. It does the artist little good to spend a lot of time building advanced models with complex materials and mapping if the engine doesn't support those features. That is one reason that this book has not dealt with things like layered shaders, bump maps, or complex modeling systems. Although some of the high-end game engines might support these features, most do not.

Maya is a very popular 3D software program and is supported by almost all game engines for exporting data. Because there are so many types of game engines, I can't cover every one in this book. However, I can show you how most exporters work. Maya supports a number of exporters for character animation. You can see a list of the exporters if you go to the Plug-in Manager (see Figure 10.30), found in the Windows directory under Settings/Preferences.

The checkboxes in the Plug-in Manager indicate what plug-ins are loaded when you run Maya. Most game engines come with an exporter that is a Maya plug-in.

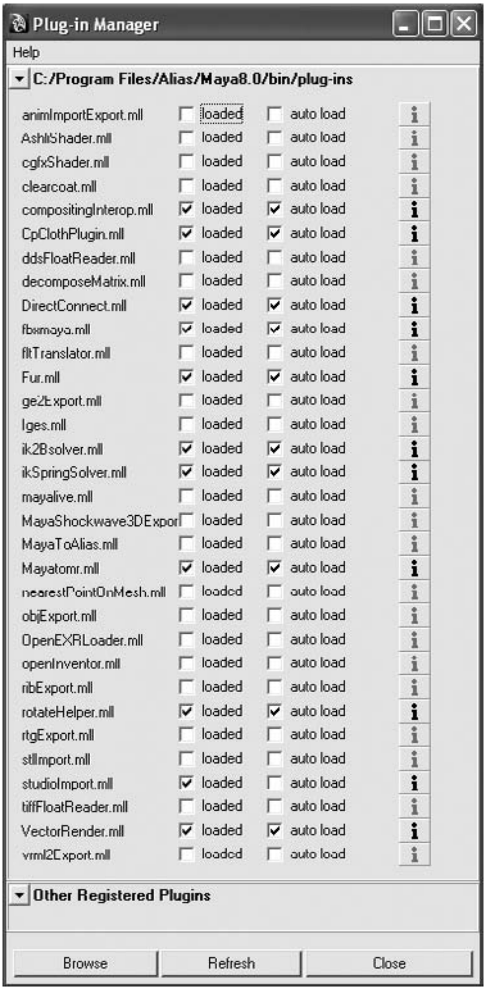


Figure 10.30
Bring up the Plug-in Manager.

You will need to follow the directions for installing the plug-in that is supplied with the engine into Maya. Once you have installed the plug-in, load Maya and check the boxes for that plug-in in the Plug-in Manager. When the exporter plug-in is active, you can export by opening the File menu and choosing Export All. This opens the Export All Options dialog box, shown in Figure 10.31.

The Export All Options dialog box has the available exporters in a drop-down list, in Figure 10.31. All you have to do is select the exporter for your engine, select from the options available in your exporter, and click Apply to export your animation to for the engine.

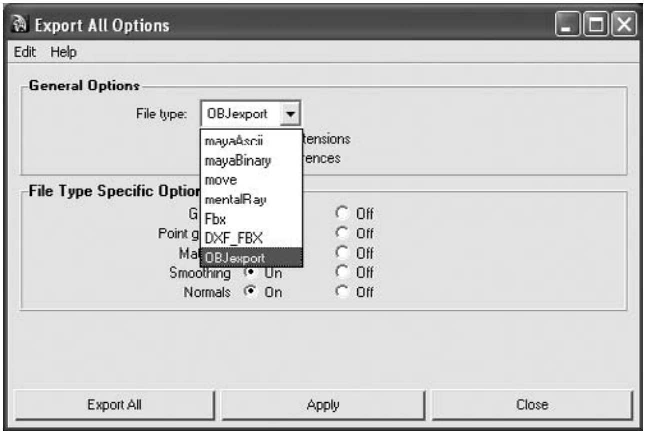


Figure 10.31
You can use Maya's Export feature to export your models.

Summary

This chapter introduced you to the exciting world of non-linear animation and touched briefly on exporting your animation to a game engine. It introduced you the Trax editor and explained some of its features.

In this chapter you learned the following:

- Creating a character set
- Creating a clip
- Creating a pose
- Enabling and disabling clips
- Moving clips around in the Trax editor
- Applying poses
- Using the Plug-in Manager
- Exporting character animations

All game engines do not support non-linear animation, but some do. Check with the documentation that came with your game engine and see if it supports non-linear animation.

Closing Remarks

This has been a whirlwind tour of character animation for games. I have only touched on the many processes and animation features in Maya and other art programs. I hope that what you have learned from this book will help you to further explore game character animation. Good luck in your work. I hope to see some great character animation from you in the future.

If you would like a copy of the model and animation that was created for this book, go to http://www.turbosquid.com and look up Warrior by Les Pardew.

INDEX

D

E

F

G

H

I

J–K

L

M

N

O

P

Q–R

U

V

W–Z

This page intentionally left blank

License Agreement/Notice of Limited Warranty

By opening the sealed disc container in this book, you agree to the following terms and conditions. If, upon reading the following license agreement and notice of limited warranty, you cannot agree to the terms and conditions set forth, return the unused book with unopened disc to the place where you purchased it for a refund.

License

The enclosed software is copyrighted by the copyright holder(s) indicated on the software disc. You are licensed to copy the software onto a single computer for use by a single user and to a backup disc. You may not reproduce, make copies, or distribute copies or rent or lease the software in whole or in part, except with written permission of the copyright holder(s). You may transfer the enclosed disc only together with this license, and only if you destroy all other copies of the software and the transferee agrees to the terms of the license. You may not decompile, reverse assemble, or reverse engineer the software.

Notice of Limited Warranty

The enclosed disc is warranted by Thomson Course Technology PTR to be free of physical defects in materials and workmanship for a period of sixty (60) days from end user's purchase of the book/disc combination. During the sixty-day term of the limited warranty, Thomson Course Technology PTR will provide a replacement disc upon the return of a defective disc.

Limited Liability

THE SOLE REMEDY FOR BREACH OF THIS LIMITED WARRANTY SHALL CONSIST ENTIRELY OF REPLACEMENT OF THE DEFECTIVE DISC. IN NO EVENT SHALL THOMSON COURSE TECHNOLOGY PTR OR THE AUTHOR BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING LOSS OR CORRUPTION OF DATA, CHANGES IN THE FUNCTIONAL CHARACTERISTICS OF THE HARDWARE OR OPERATING SYSTEM, DELETERIOUS INTERACTION WITH OTHER SOFTWARE, OR ANY OTHER SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES THAT MAY ARISE, EVEN IF THOMSON COURSE TECHNOLOGY PTR AND/OR THE AUTHOR HAS PREVIOUSLY BEEN NOTIFIED THAT THE POSSIBILITY OF SUCH DAMAGES EXISTS.

Disclaimer of Warranties

THOMSON COURSE TECHNOLOGY PTR AND THE AUTHOR SPECIFICALLY DISCLAIM ANY AND ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY, SUITABILITY TO A PARTICULAR TASK OR PURPOSE, OR FREEDOM FROM ERRORS. SOME STATES DO NOT ALLOW FOR EXCLUSION OF IMPLIED WARRANTIES OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THESE LIMITATIONS MIGHT NOT APPLY TO YOU.

Other

This Agreement is governed by the laws of the State of Massachusetts without regard to choice of law principles. The United Convention of Contracts for the International Sale of Goods is specifically disclaimed. This Agreement constitutes the entire agreement between you and Thomson Course Technology PTR regarding use of the software.