Chapter 1

Modifying the Game

In This Chapter

- Looking at the game through a modder's eyes
- Finding modding tools that you had all along
- Walking through the making of a mod
- Going public with your creations

Have you ever been playing a video game and thought, "I would have done it differently" or "I could have done it better"? Perhaps you thought, "Wouldn't it be cool if. . . . " Well, you don't have to just think it. You can make changes to games, and you don't have to be a software engineer to do it.

Game modification — changing something in a game — has generally been associated with the first-person shooter (FPS) and real-time strategy genres. The change could be very small, such as making a player's outfit orange instead of blue, or the change could be very large, such as creating a whole new environment for the player to explore. You can change almost every aspect of a game and make it look and feel like something completely different. Or, instead of altering an existing part of the game, you could add new elements to it. Anything that in some way modifies a game from what it was when the publisher released it is a *mod*.

Game modification isn't a new practice. However, only recently, with the creation of multiplayer shooters for the PC, has it become popular. It was this genre of gaming that gave people the inspiration to show off. At first, players competed to see who was the best FPS player. Later, when players realized that they could modify a game, the competition grew to include this aspect of the game and to see who could make the most impressive changes to a game.

The FPS game genre was created in early 1990. You play from the point of the view of the in-game character just like in *Half-Life 2*. This gives you your first-person perspective of the game.

COPYRIGHTED MATERIAL

The center of action revolves around you, *Gordon Freeman,* as the player, while you use a handheld weapon, such as a pistol. Although your primary objective in the game might change, you're often placed in situations where you must shoot your weapon — thus, making the game a *shooter.*

Controversy surrounds which game was the first FPS game. It's a tossup between *Spasim* and *Maze War,* which were first developed in 1973. Then, later that same year, player versus player game play was tested between two linked computers playing *Spasim.* The following year, both games were introduced to a network, and multiplayer gaming as we know it was invented. Because both games played from the first-person perspective with weapons, this marked the birth of the FPS.

In 1991, id Software released the game *Hovertank 3D,* which was a simple maze game from the first-person perspective. The environment was very flat, and the enemies were nothing more than 2D graphics. Later that year, *Catacomb 3D* was released (as a modified version of *Hovertank 3D*) featuring textured walls as well as showing the player's hand onscreen — like you now see in *Half-Life 2.*

In 1992,VGA (Video Graphics Array) graphics was added to the release of *Wolfenstein 3D.* This game was a huge hit due to both the game play and this higher-quality graphics and inspired more development in the genre. The following year, *Doom* added even more graphical detail. This game offers rooms of various sizes, outdoor environments, and textures that were previously flat surfaces. However, the most important upgrade to this rising game genre was the ability for anyone connected to a network to enjoy the multiplayer aspect.

The first version of *Quake* was introduced by id Software in 1996. It had highly upgraded graphics as well as networking capabilities and was the first game in the genre to gain widespread fame as a multiplayer Internet game. It broke the bounds of its predecessor, *Doom,* by networking globally. To further its success, *Quake* was the first game that offered developer support for user modifications. This was the beginning of mods created by the consumer rather than the industry, and it was also when Valve Software began work on its game, *Half-Life.*

In 1998, Sierra Studios and Valve Software released *Half-Life.* This FPS was based on a heavily modified version of the *Quake* game engine. At first, what made this game a huge hit was the presentation of the game and that it had a storyline with a plot. Not many games before it had actual plots to involve the player; rather, they simply offered the player something to shoot. For the first time, there is a game presenting the player with an interesting story.

Developed on an adaptable game engine, *Half-Life* continued to encourage the gaming and modding community to further develop *Half-Life.* Valve Software provided excellent support from the beginning by including level-design tools with the software. Later, Valve Software released documentation, additional source codes, and tools to further the capabilities of modders.

Then, in 1999, a beta version of *Counter-Strike,* a *Half-Life* mod, was released to the public. This mod grew in popularity like no other mod before it. It was so well received that Valve bought the rights, assisted in its continued development, and released it as a commercial expansion of the game in 2000.

Many other mods were created for *Half-Life*, but *Counter-Strike* set the standard. It was proof that modding could get you into the gaming industry and benefit the game developers.

Valve Software took five of the six years since the release of *Half-Life* to deliver a game of which they were proud. That game is *Half-Life 2* and it has been redeveloped from the ground up to provide us with the best possible game play to date. Now we, as Gordon Freeman, can continue the story where it left off.

Checking Out Half-Life 2

Whether you purchase *Half-Life 2* with or without the intent to modify it, you should begin by playing around with it. Play through a few single-player missions and then move on to the multiplayer games. If you don't want to play online, start multiplayer games of your own. You might be the only one in the game, but you will still enjoy yourself.

After playing the game and enjoying what the developers were able to deliver, play the game again — but this time, instead of running around and shooting everything that moves, take some time to look around. Stop and look out windows and over railings. Walk around the other players in the game and see what they're wearing. Take a closer look at the walls to see the details that are included, and then see what happens when you shoot them with different weapons. Listen to the sounds the weapons make as well as the sounds all around you.

By investigating the details of the game, you start to see things differently. It's like looking at a room where you live and thinking about painting the walls a different color or moving the furniture. It could also be like considering a different outfit for the day as opposed to the same outfits that you wear every day.

Adding to or Changing the Game

At first, seeing which game elements you can change can be difficult. However, when you begin to understand the different pieces that make up the game, you will start looking at all games a little differently. You can relate the various elements you see to specific files within the game, and you'll start to know which of those files that you can modify. For instance, look around the physical area in which you're now sitting. Within the area, you see objects, like the book you're holding; a table with some items on it — or, if you're outdoors, maybe some trees. In the game world, each different item could be considered a separate object that the game refers to as an *asset.* Each asset, because it may be used more than once in a game, is defined in files. If you change one of these files — say, to change the book you're holding to a different color — you made a modification.

So what does this have to do with mods and modding? Well, if you modify the game so that it's in any way different from when you purchased it, you create a mod. *Mod* is just a short way of saying *modification.* Then, it stands to reason that the act of modifying is called *modding.*

The mods that you make can be simple or complex. You can make them by adding something new to the game or by changing something that already exists. You could make your changes to provide an improvement to the game, or you could completely change everything and create a *total modification* of the game. You might be surprised to know that many of the games on the shelves are total modifications of another game. The original *Half-Life* game is essentially a total modification of *Quake* from id Software, and *Half-Life 2* was built from the basis of *Half-Life.*

Finding out what you can mod

Games are just groups of files that are read by one master program that displays those files' contents on the screen. When one or more of these files is changed, the change is reflected within the game. Official game updates and expansion kits can perform these changes, but you can, too. So why not include your own changes to the game to create something completely new?

Upon first glance, you might not realize just how much game content you can mod. Everything, all of what you see from the time you double-click the game icon to the moment you close down the game, can be changed. A short list of moddable things in a game would read like this:

- **Textures and images:** Everything that you see when you play any level in *Half-Life 2* started as an image. Whether it's the bricks on the walls or the face on another player, these are all images that can be added. I show you in Chapter 13 how to create your own textures.
- **Levels:** From multiplayer to single player, you can build completely original levels for the game that you and your friends can play. What could be more fun than playing a multiplayer level together with your friends online?
- **User interfaces:** The selection windows before playing the game and the usable computer screens within the game can be modified. You can set up these screens to better meet your needs or to make things look any way you have dreamed.
- **Coding changes:** Change the way a weapon shoots or how much damage a player receives when they fall from the top of a building. This and much more can be achieved through code changes and *Valve Software* has given us permission to make them.

The preceding is just a very short list of what can be modded in this game. As long as you have access to the files that make a game run and you have the tools to change them, you can modify that game as much as you like. You could even turn *Half-Life 2* into a new version of *Donkey Kong* if that's what you want.

The reason *Half-Life 2* can be modded so extensively is primarily due to the developers. Luckily, players have been provided with access to the game files so that we can modify them. Not all game companies permit that kind of access.

Knowing what tools you need

Tools are available for every job, and game modification is no exception. Some tools are provided for you by the game developers, but others you must obtain. However, you might be surprised to know that most of what you need, if not all, is already installed on your computer. You just need to know which programs you can use to modify each of the different files within the game.

Here are examples of such programs that you might already have:

- **File-compression utility:** If you've been downloading a lot of files, you're certain to have a file-compression utility. The program of choice in this book is WinZip, which is used primarily to open `.zip` files found on most download sites. This utility can easily open the compressed game files that were installed with the game so that you can gain access to all the moddable game files and start having fun.

- **Plain text editor:** Many of the files in the game are written and modified by using a plain text editor. I'm certain that you already have at least one of these installed on your computer. Notepad, for instance, is a perfect program for editing these files. It comes installed with Windows and can read, edit, and save these files without any special setup or instruction.
- **Developer-provided tools:** When you install the *Half-Life 2* game, you also have the option to install some of the modification tools via the Steam engine, *Source SDK.* This software, a developer's kit for the Source engine, allows you to modify the game and create custom game levels to play.

As modding became popular, game developers started to assist the modification community. They offered words of advice and eventually tools and documentation to make more complicated changes. As the modifications became bigger and better, so did the sales of the original game because more and more people wanted to play the game with these new modifications installed. This inspired more participation from developers and publishers who offered even better tools and documentation.

The Modding Process Goes Something Like This

The most common type of modification is to create a custom level for the game. The process of doing such goes like this:

1. **Plan your custom level with notes and drawings.**

   Write down what you want to include in your custom level and maybe even sketch out how you want things to lay in the game.

2. **Construct the level in a program by building walls.**

   This is a lot like playing with blocks. You create and place your different shaped blocks where you want them in order to create a room, several rooms, or any other structure for the player to roam.

3. **Add some color to the surfaces in the level.**

   Adding color is a simple process of selecting an image and then applying color to the wall, floor, or any other surface in the game.

4. **Place additional elements in the game such as lights, monsters, weapons, or other objects.**

Again, just select elements from a list and place them where you want. Then you can fine-tune the way they work. (For instance, you can change the color of a light.)

5. **Compile and play the level in the game.**

   Choose a compile command from the editor's menu, and it creates all the files required so that you can play your finished level in the game. Then you just load the level and start having fun.

6. **(Optional) Give your level to the world.**

   This optional step puts all the custom files together into a single file that you can place online for download or on a disc to hand to your friends. This way more people can enjoy the work you put into your custom modification.

As you can see, the process isn't all that complicated. In this book, I show you where you can find the necessary tools, how to use them, and the options that each tool has to offer. With this information, you soon will be on your way to making your own custom game levels.

Sharing the Game with Others

In the list in the preceding section, I mention that the last step of the modification process is optional; however, sharing your creation is most often the purpose of making a mod. I think that it's perhaps the most exciting part. For my part, knowing that many other people are getting enjoyment from something that I built motivates me to do more.

In this book, I not only show you how to package all your files together for distribution, but I also show you where to go from there. I offer advice on where to send your files and how to get them out to the public for all to enjoy.