

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 is the process of changing something in a game and has generally been associated with the first-person shooter 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 could 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 termed a *mod*.



Game modification is not a new practice. It has been going on for quite some time, but 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 first-person shooter (FPS) player. Later, when players realized that they could modify the game, the competition grew to include this aspect of the game and to see who could make the most impressive changes to a game.

Shoot First, Ask Questions Later

The FPS game genre was created in early 1990. You play from the point of view of the in-game character just like that of *Quake 4*. Also, like *Quake 4*, it is id Software, Inc., with the creation of *Doom*, that made this genre popular.

As far as staking claim to being the first FPS game, there is much controversy. It's a toss up between *Spasim* and *Maze War*, which were first developed in 1973. Then, later that 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 you 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 *Hovortank 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 this *Hovortank 3D*. *Catacomb 3D* displayed textured walls and showed the player's hand on-screen like you now see in *Quake 4*.

1992 offered the addition of VGA graphics with the release of *Wolfenstein 3D*. This game was a huge hit and inspired more development in the genre. The following year, *Doom* added even more graphical detail. This game offered rooms of various sizes, outdoor environments, and textures on what 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.

From that point on, the FPS genre grew. Graphics greatly improved over time as innovations in computer hardware developed. The ability to process more information has increased the number of simultaneous events that can happen during a game. All the while, several new games emerged that attempted to copy the success that id Software started. *Quake* and *Doom* sequels were produced, bringing further enhancements. *Quake 4* is the legacy of a great series of games from a great developer, id Software.

Checking Out Quake 4

Whether you purchase *Quake 4* 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 were 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 will start to see things differently. It will be like looking at a room where you live and thinking about painting the walls a different color or moving the furniture around. It could also be like considering a different outfit for the day as opposed to the same green armor you wear every day. (You're a *Quake 4* marine, remember?)

Adding To or Changing the Game

At first, it can be difficult to see which game elements you can change. However, as you begin to understand the different pieces that make up the game, you will start looking at all games a little differently. You will be able to relate the various elements you see to specific files within the game, and you'll start to know which of those files you can modify.

For instance, look around the physical area in which you are now sitting. Within the area, you see objects like the book you are now holding; a table with some items on it; or if you are outdoors, maybe some trees. In the game world, each different item could be considered a separate object that the game would refer 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 so that, say, the book you are holding is a different color, you have just 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 have

created a mod. *Mod* is just a short way of saying *modification*. Then, it stands to reason that the act of modifying could be called *modding*.

The mods 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 what is called 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. *Quake 4* is essentially a total modification of *Doom 3*. There are quite a few additions to the code of the game, but Raven Software, Inc., based the *Quake 4* game on the *Doom 3* game developed by id Software.

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:

- ✔ **Loading screens:** The loading screen for the game and the loading screens for each level of the game. You can change all these screens from their original designs. Right now the program's loading screens show a lot of green and the primary *Quake 4* symbol, but you can turn these screens into something else if you want. In Chapter 13, you see how to create a custom loading screen that is displayed while loading your level.
- ✔ **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. Chapter 16 shows you how you create your own user interface and use it within a level.
- ✔ **Textures and images:** Everything that you see while playing any of the levels within *Quake 4* 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. Chapters 15 and 18 show you how to do exactly this.

- ✓ **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?

The preceding is just a very short list of what you can mod 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 *Quake 4* into a new version of *Donkey Kong* if that's what you want.

The reason *Quake 4* can be modded so extensively is primarily due to the developers. Luckily, id Software has provided access to the files of the game so that you can modify them. Not all game companies do that.

Knowing what tools you need

There are tools 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.

As you dig deeper into modding, you will need to install additional tools that you might not yet have. However, these tools have been supplied for you on the CD-ROM located in the back of the book. For the most part, these tools are for editing images that are used in the game and will not require any foreknowledge of their use. When it comes time to use these tools, you will be instructed on what to do and how to do it.

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 applying it 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 them. 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 out there 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.

