

SAMS

# MASTERING UNREAL TECHNOLOGY

The Art of Level Design



Welcome to  
**MASTERING UNREAL TECHNOLOGY:**  
The Art of Level Design

Welcome to **Mastering Unreal Technology: The Art of Level Design**. You're going to take an exciting journey through the rich world of game design, using the Unreal Engine. This book is dedicated to the aesthetic aspects of game design, such as level design creation, character development, and static mesh design. However, this book does not cover many game design topics from the programmer's perspective, meaning there isn't much emphasis on programming the Unreal Engine through its native programming language: UnrealScript.

This book has been designed for a wide range of people, actually. The first group are aspiring mod makers or fledgling game designers. This book is also aimed at anyone who has ever wondered exactly what kind of work goes into making his or her favorite game.

Beyond gamers, however, this book can be a great resource for those in the fields of architecture or construction visualization. With the Unreal Engine, you can offer your clients real-time flythroughs and simulations that they can view and explore. How to apply the Unreal technology that drives today's cutting-edge games is limited only by your imagination.

This book is designed to be read from start to finish, although this isn't necessarily the best way for everyone to complete this book. You'll find that each chapter has a robust selection of information about each relative topic; however, the book has been designed to be flexible enough so that you can start with the areas of game design that interest you the most.

Whatever your motive, this book is your resource, so use it as you will. However, keep in mind that game design is a interconnected field, in that many of its aspects are directly connected to each other. You might not want to be a level designer or static mesh constructor, but you'll make yourself far more valuable to a prospective employer (or mod team) by knowing about each field of game design.

## **How This Book Is Organized**

### **Part I: The Unreal Universe**

This first part of the book focuses on a general introduction to level design for the Unreal Engine with its native editing system: UnrealEd. It begins with a brief history of Unreal and quickly moves on to a thorough introduction to how the Unreal Engine functions.

From there, Part I walks you through creating your first Unreal Level. You see how to carve your own environments into the Unreal world, how to texture and light them, and how to add the elements that make them aesthetically pleasing and still fun to play. The topics covered include creating indoor and vast outdoor environments, creating and controlling your level's physics through the use of volumes, and generating a variety of lighting effects. Part I wraps up with an in-depth look at material and texture creation and how to create interactive level elements, such as moving doors and controllable elevators.

## **Part II: Advanced Design Techniques**

Part II opens the doors to more high-end features of the Unreal Engine. Beginning with an introduction to working with real-time particle systems and effects, Part II also brings you face to face with the Karma physics engine. It explains how it can be used to create realistic simulations based on the laws of physics. Next, the chapter covers the techniques for controlling the behavior of Unreal's artificially intelligent players known as "bots."

Part II then moves on to cover Matinee, UnrealEd's integrated system for creating in-game cinematics—animated clips and films—with the game engine. Part II then discusses how to create scripted in-game sequences to create exciting events that pull players deeper into the action. This part closes with a look at optimizing your levels to keep gameplay as fast as possible, while pulling as much power as you can from the engine.

## **Part III: External Design**

Part III introduces you to creating 3D game elements with Alias's Maya. This part begins with an introduction to Maya from the beginner's perspective. You're quickly brought up to speed on navigating the software, learning techniques for polygon modeling, creating static meshes for use as decorative elements in your levels, and preparing your models for the application of texture.

Next, Part III explains techniques for creating full-character models. It then covers how you can turn that model into a digital puppet so that it can be animated and eventually used as a playable character in your Unreal game.

## **Special Features**

This book contains a series of in-depth tutorials, outlining each major concept introduced. To assist you in following these tutorials, the book includes a DVD packed with all the necessary assets, the Unreal Runtime Engine, and Maya Personal Learning Edition 5.0. The book also includes an appendix that serves as a user's guide for UnrealEd and another that lists the pertinent properties for Unreal's most important in-game assets.

## **Book Ordering Information**

### **Mastering Unreal Technology: The Art of Level Design**

By Jason Busby, Zak Parrish, and Joel VanEenwyk

Authoritative guide to building custom mods, maps and levels with the Unreal engine from Epic Games.

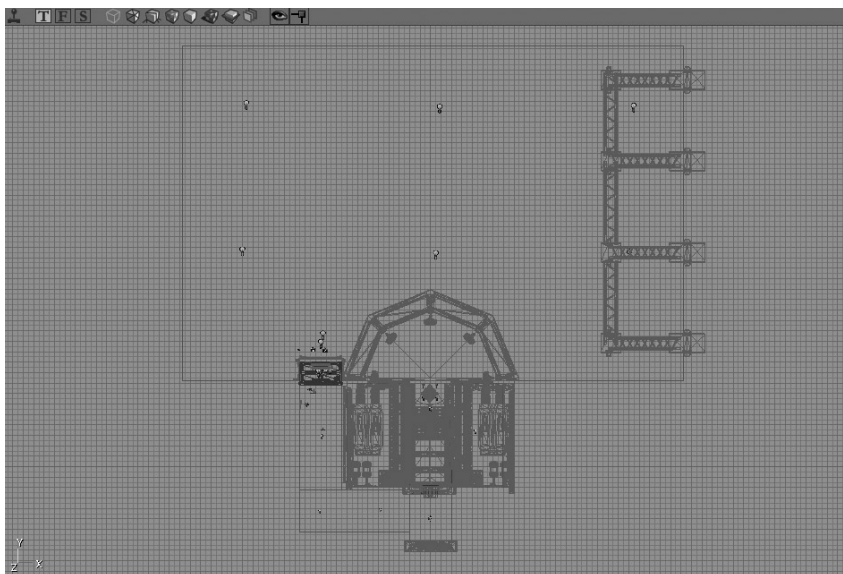
ISBN: **0-672-32692-2** • PRICE: **\$49.99 US** • IN-STOCK DATE: **NOV. 2004**

**Available at [Amazon.com](http://Amazon.com)**

## Creating Your First Map

Now that you know all these important aspects of level design, it's time to do some real level construction. The following tutorials introduce you to a variety of fundamental concepts and tasks that you will need to understand before you're ready to complete your own levels. You'll begin simply by subtracting out a box-shaped room, and then "paint" it with texture, add some lights, and go on from there. **FIGURE 3.21** shows the completed level you will end up with after completing the tutorials in this chapter.

3




**FIGURE 3.21** A completed level.

### Subtracting the First Room

**TUTORIAL 3.2** shows you what you need to know to create your first open space from Unreal's world of infinite solid mass:

#### **TUTORIAL 3.2:** Subtracting a Simple Room

1. Start UnrealEd and begin a new scene by choosing File > New from the menu.
2. You need to define the shape and size of your Builder brush so that you can create a subtractive brush of the correct dimensions. Open the CubeBuilder dialog box by right-clicking on the Cube primitive  in the CSG section of the toolbox.



3. In the CubeBuilder dialog box, enter the following settings (see **FIGURE 3.22**), and then click the Build button. When you do, you'll notice that you can see the Builder brush as a red cube of the dimensions you selected. You can then close this dialog box.


Height: 384

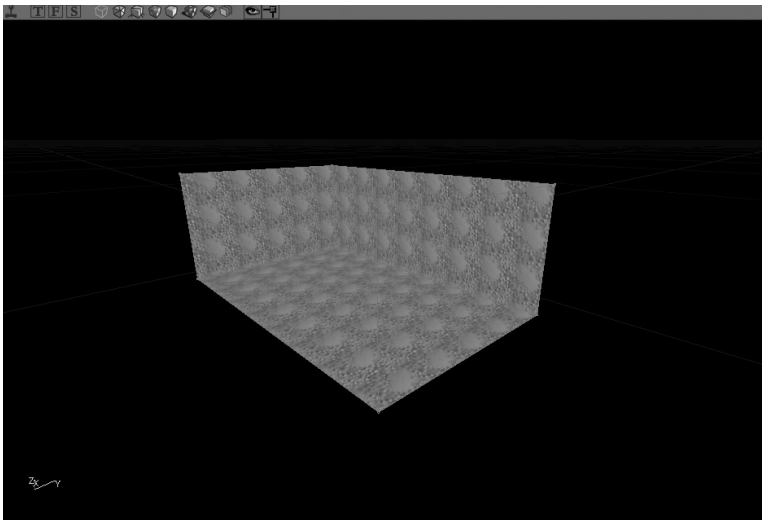
Width: 636

Breadth: 1024



**FIGURE 3.22** Dimension settings in the CubeBuilder dialog box.

4. Click the Subtract button  in the toolbox. You have just created the first subtracted space of your map (see **FIGURE 3.23**).



**FIGURE 3.23** A subtracted room.

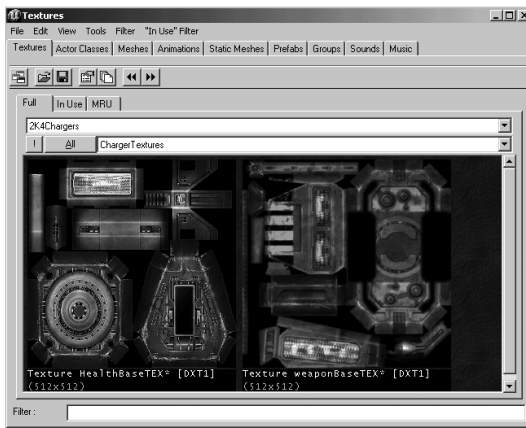
## END TUTORIAL 3.2

### Adding Textures

Right now, the textures on your room's surfaces look rather bland because they're covered with the default texture. If you don't select a texture from the Texture browser before adding or subtracting mass in your map, the surfaces of the resulting brush are covered with the default

texture—a greenish bubbly surface. Selecting a texture from the Texture browser *before* adding or subtracting is a good idea. If you forget to do this, you can still change those textures by using the Texture browser.

To open the Texture browser, click its icon on the toolbar of the UnrealEd window. You can use this browser to view textures included in texture packages and to filter the number of displayed packages by selecting specific groups (see **FIGURE 3.24**). You can find out more about the menus and features of the Texture browser in Appendix A.

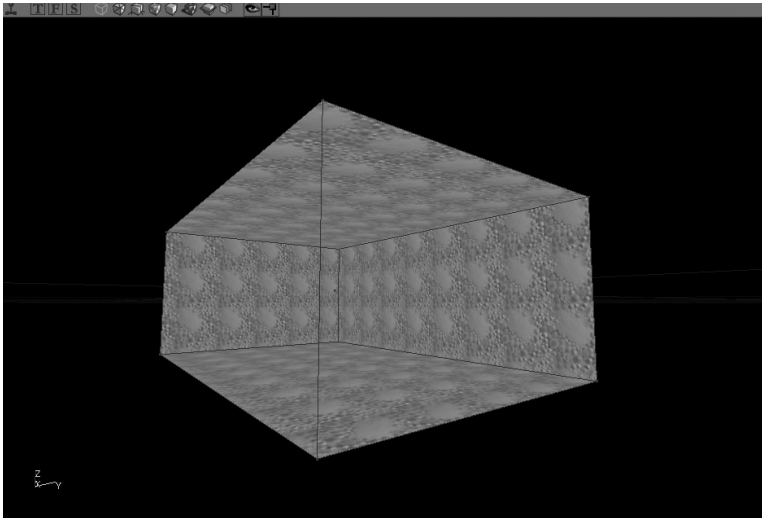


**FIGURE 3.24** The Texture browser.

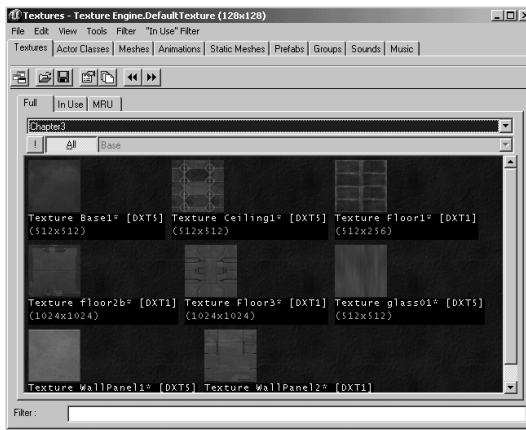
**TUTORIAL 3.3** demonstrates how to select textures for the walls, floor, and ceiling of your room. The textures you need are stored in a package, so first you need to load that package.

### TUTORIAL 3.3: Texturing the Room

1. Notice that if you click on any surface in your room, it turns blue to indicate it is selected. Before you begin texturing, make sure no surfaces are selected (see **FIGURE 3.25**). If a surface is selected, click it again to deselect it.
2. Open the Texture browser using the method described at the beginning of this section.
3. From the browser's menu bar, choose File > Open and select the **Chapter3.utx** file (see **FIGURE 3.26**). This package contains several textures divided into the following groups: Base, Ceilings, Floors, Wall, and Glass. You can switch between these groups by using the drop-down list next to the All button. Click the All button to see all textures in the package, regardless of what group they're in.



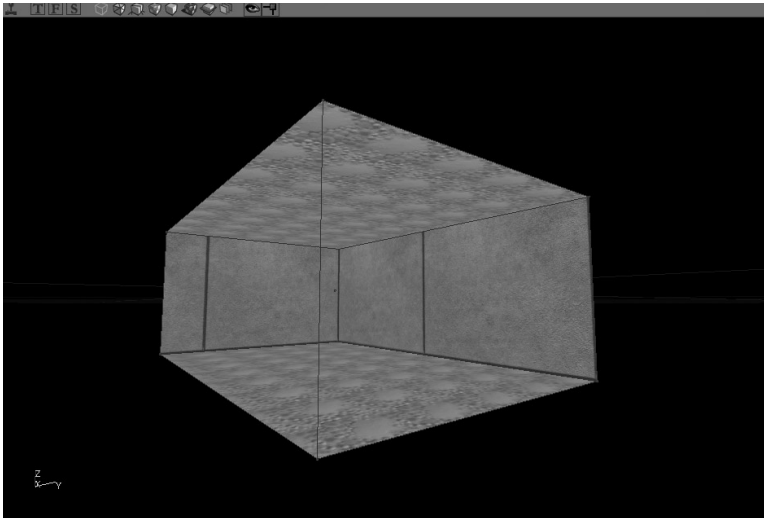
**FIGURE 3.25** The room before texturing.



**FIGURE 3.26** The Texture browser with the Chapter3.utx file loaded.

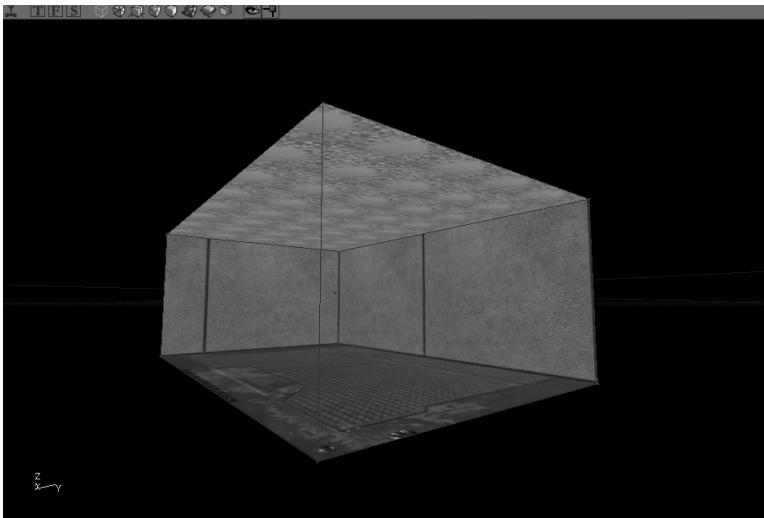
4. Click the list arrow next to the Add button, select the Walls group, and click the **WallPanel1** texture. The selected texture isn't applied to any surfaces in the level because no surfaces are selected.
5. Select the four walls in the room by Ctrl-clicking each one in the 3D viewport.
6. Click the **WallPanel1** texture again in the Texture browser. Notice that the texture is then applied to all walls (see **FIGURE 3.27**).





**FIGURE 3.27** Walls with the WallPanel1 texture applied.

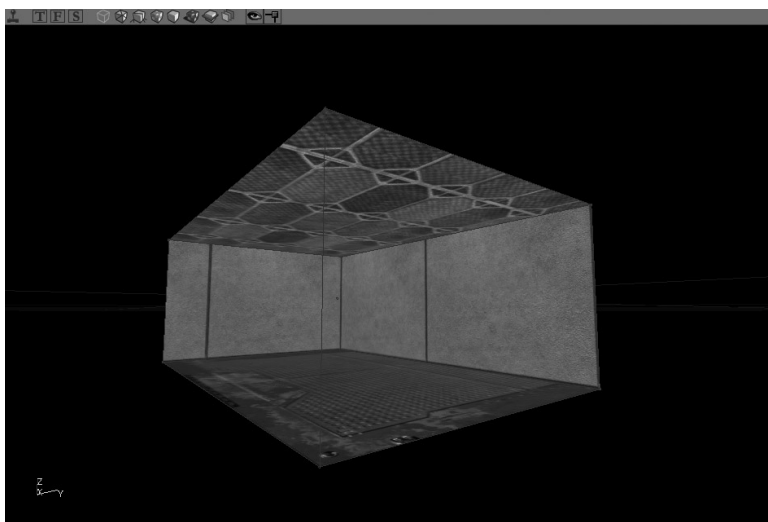
7. In the 3D viewport, select the floor surface.
8. In the Texture browser, select the Floors group and click the Floor2b texture (see **FIGURE 3.28**).



**FIGURE 3.28** The floor texture applied.

9. Use the methods in steps 7 and 8 to select the ceiling surface and apply the Ceiling1 texture (see **FIGURE 3.29**).





**FIGURE 3.29** A textured room.

### END TUTORIAL 3.3

---

## Texture Fitting

You have now applied textures to your level, but they don't seem to be placed well. Using the Surface Properties window, you can adjust how the textures appear on each surface by using the surface's UV coordinates. A surface's UVs are a two-dimensional coordinate system for controlling the placement of a texture on a surface. In most cases, U is in the horizontal direction, and V is in the vertical.

You can type numeric values in the Surface Properties window instead of adjusting these coordinates interactively. The Surface Properties window also gives you access to special alignment tools not found in the toolbox. In **TUTORIAL 3.4**, you take a look at how to adjust textures in your level. Note that many of the numeric values given in this tutorial were derived through experimentation.

### TUTORIAL 3.4: Fitting the Textures

---

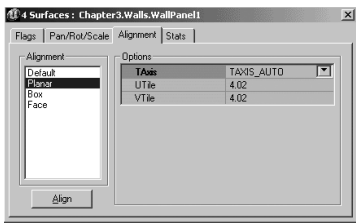
1. Select the four walls in the room.
2. Click the Surface Properties button or press F5.

3.
- In the Surface Properties window, select the Alignment tab, set the following options (see **FIGURE 3.30**), and then click the Align button:

Alignment (list box at the left side of the window): Planar

UTile: 4.02

VTile: 4.02

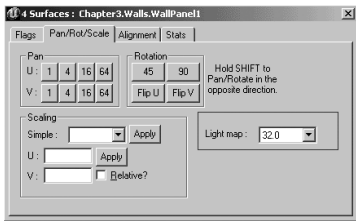


**FIGURE 3.30** Alignment tab of the Surface Properties window.

**NOTE**

You can pan in V in a downward direction simply by clicking any of the incremental buttons. Pressing Shift while clicking pans the texture back up. This is, of course, in relation to the surface you’re facing.

4.
- The wall appears to have tiles, but the alignment is still off because the tiles at the edges of each surface look cut off. To fix this problem, in the Surface Properties window, select the Pan/Rot/Scale tab (see **FIGURE 3.31**).
5.
- As you can see, the textures need to be moved vertically to align correctly with the floor and ceiling. With all four walls still selected, click the Pan V 64 button four times. You can see the textures move downward 64 units each time you click the button. Four clicks should make everything align with the top and bottom of the walls.
6.
- You’re almost finished, but you can see that the two short walls need their textures aligned horizontally. Go ahead and select these two walls now.
7.
- Click the Pan U 64 button four times. The walls should then be aligned correctly (see **FIGURE 3.32**).
8.
- For the ceiling and floor alignments, the following are the recommended alignment numbers. Using the technique demonstrated earlier in this tutorial, see if you can get everything aligned correctly. **FIGURE 3.33** shows an example.



**FIGURE 3.31** Pan/Rot/Scale tab of the Surface Properties window.

**NOTE**

If all four walls are still selected, you can simply Ctrl-click the two longer walls, thereby deselecting them.

**Floor**

Alignment: Planar

UTile: 8.0

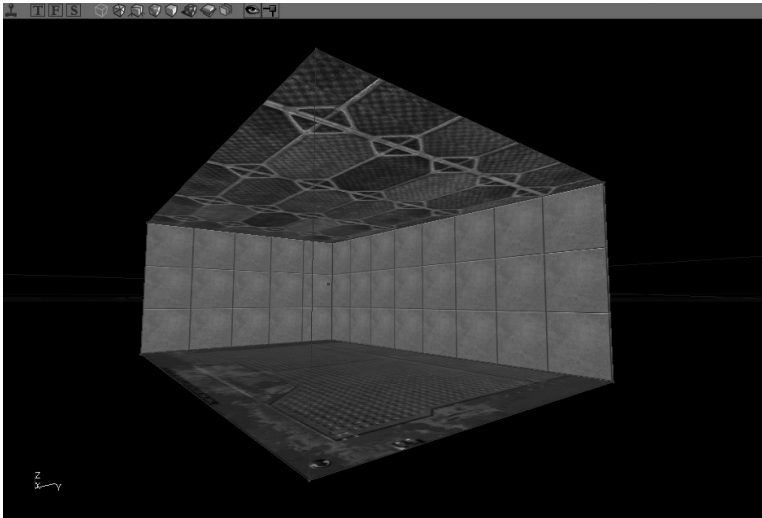
VTile: 8.0

**Ceiling**

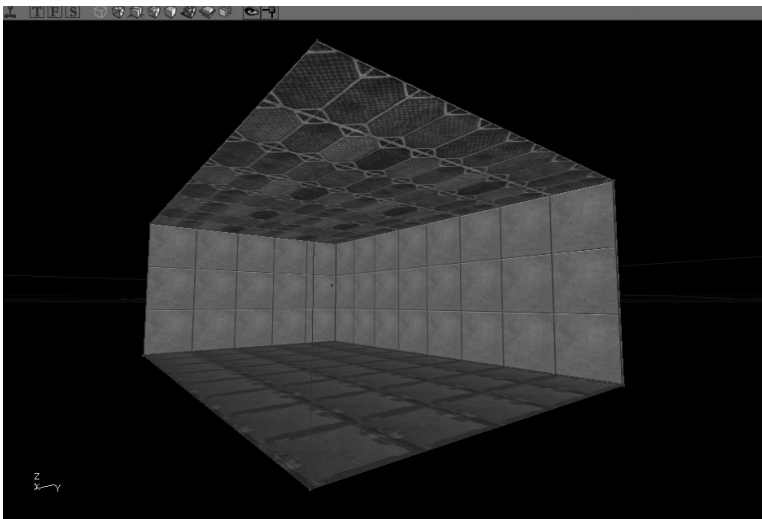
Alignment: Planar

UTile: 2.0

VTile: 2.0



**FIGURE 3.32** Correctly aligned walls.



**FIGURE 3.33** Correctly aligned floor and ceiling.

#### END TUTORIAL 3.4

---

## Making the Level Functional

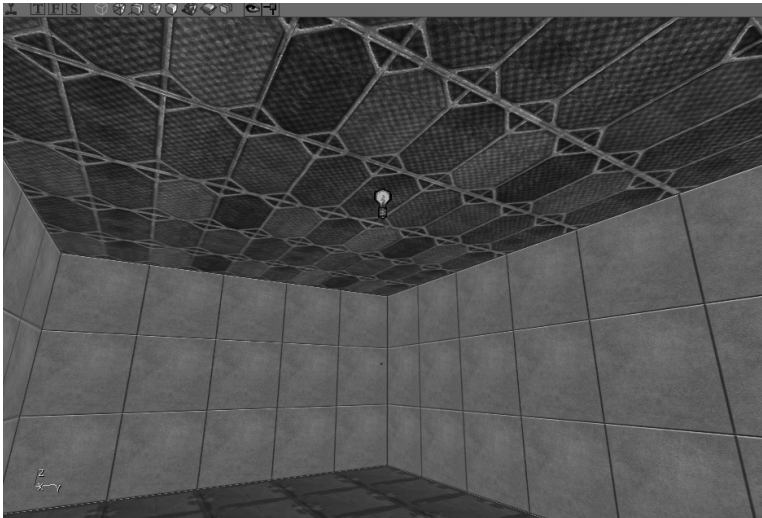
Lights, as you might imagine, are essential to your levels. Lightless game levels look much like a room with the lights turned off. Obviously, playing a lightless game would be quite difficult. Lights are used not only to allow players to *see* your level, but also to create mood, add color, and control the dramatic “feel” of each area of the map. In the following tutorial, you make your level playable by learning how to place lights and rebuild the level to generate light maps for creating areas of light and shadow.

### 3

Before you can jump in and actually run around this map in the game, you need to add a few missing items. The first item, as mentioned, is a light source of some kind. The second is a Player Start Actor, which gives your map a point of entry for any characters that want to play it. Finally, you must rebuild the level to calculate its world geometry and generate the light maps that make surfaces seem to be illuminated.

### TUTORIAL 3.5: Making the Level Functional

1. First, add a light. Right-click on the center of the ceiling surface in the 3D viewport, and choose Add Light Here from the context menu. A light bulb icon is displayed to represent the light's position (see **FIGURE 3.34**).



**FIGURE 3.34** Newly created light.

2. At first, you probably won't see much change in the level. To see what the light will look like in the room, you must rebuild the level. Click the Rebuild All button on the toolbar.

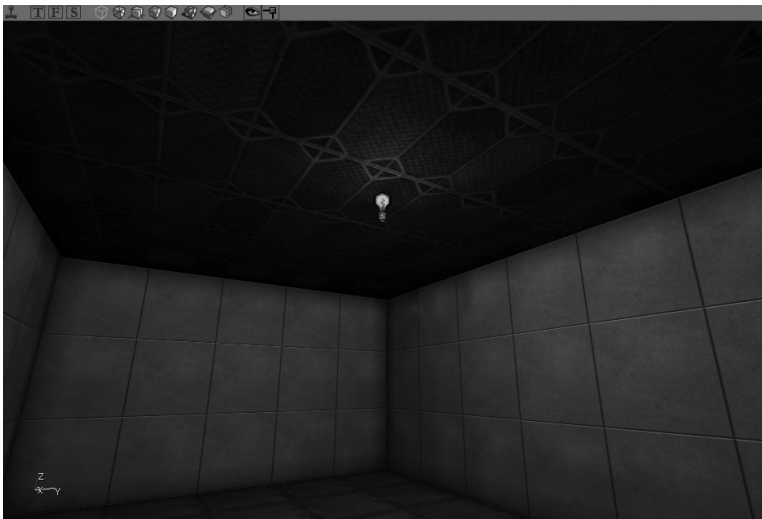
You can see some dim lighting with quite realistic softened shadows in the corners (see **FIGURE 3.35**). The lighting conditions in the room are affected by several things:

- ▶ Light position
- ▶ Light settings (brightness and color)
- ▶ Number of lights in the room
- ▶ Object placement and the shadows created by those objects

3. With the light selected, click the Actor Properties button on the toolbar or double-click the light's icon.

### CAUTION

Forgetting to rebuild before testing a level often results in your characters dying as soon as they enter the game! If you run into this problem, be sure to rebuild your level's geometry.



**FIGURE 3.35** Room with lighting.

4. In the Light Properties dialog box, expand the LightColor section and set the **LightBrightness** property to **128** (see **FIGURE 3.36**). You'll notice the room get brighter, but this isn't an accurate representation of the new value. To see an accurate representation, you need to rebuild the level again. Do so now.

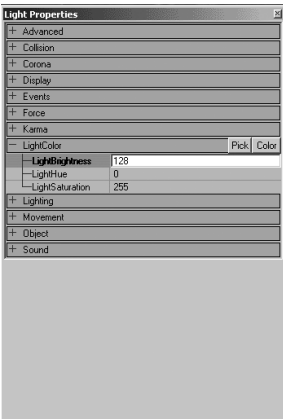
### NOTE

You can also click the Build Lights button or the Build Changed Lights button on the toolbar for the same effect.

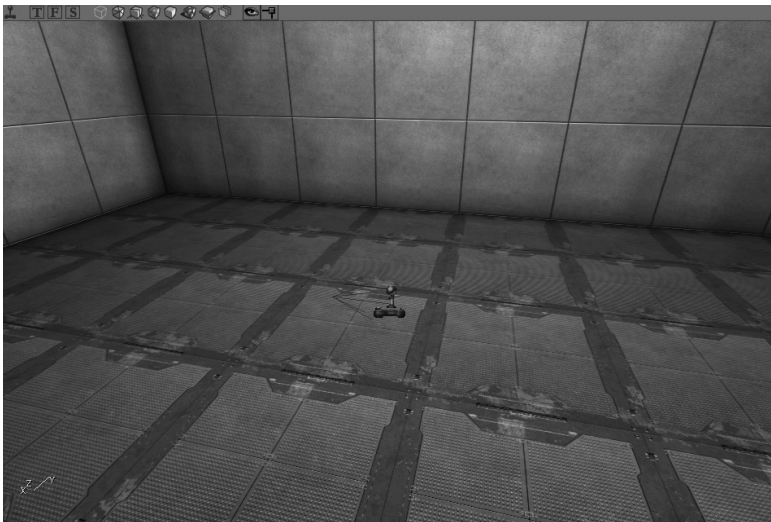
5. Now it's time to add a Player Start. Right-click on the floor surface, somewhere near its center, and choose Add Player Start Here from the menu.

A joystick icon is displayed to represent where the player starts (see **FIGURE 3.37**). When you click the icon, an arrow indicates the direction the player will be facing when he or she spawns into the map.

6. Time to play! There's no need to rebuild a level after adding a Player Start. You can simply click the Play Map! button on the toolbar and check out your new level (see **FIGURE 3.38**). Congratulations! You've finished a basic level!
7. When you're finished playing, press the tilde (~) key to close the console, and type **exit**. This immediately closes the game and sends you back to your desktop.



**FIGURE 3.36** The Light Properties dialog box.



**FIGURE 3.37** A Player Start icon with visible arrow.



**FIGURE 3.38** A completed level.

**END TUTORIAL 3.5**

---