



English

Read-me

Starting Unreal

To start Unreal:

1. Click on the "Start" button at the bottom of the screen.
2. Move the mouse to the "Programs" choice.
3. Move the mouse to the "Unreal" choice.
3. Click on the "Play Unreal" icon.

For the latest information and updates

Visit the Unreal home page on the web:

<http://www.unreal.com/>

<http://www.infogames.com/>

On our web page, you'll find:

- The latest updates & patches.
- Up-to-date troubleshooting information.
- Links to Unreal community web sites.
- The master Unreal server list.
- Links to more cool games using the Unreal technology.

Technical Support

Please see the manual accompanying the retail version of the game for instructions on obtaining technical support. See back cover for Customer Services.

TROUBLESHOOTING

Lockups

If Unreal locks up or crashes when you first run it, try starting Unreal via the "Unreal Safe Mode" icon available in the Start menu (Start/Programs/Unreal/Unreal Safe Mode). This mode is similar to Windows 95's Safe Mode. It runs Unreal with sound, DirectDraw, and 3D hardware support disabled. This way, you can modify the options in Unreal's "Advanced Options" menu that may be causing problems, then run Unreal again.

Crashes

If Unreal stops with an "Unreal has run out of virtual memory" message, you need to free up more hard disk space on your primary drive (C:) in order to play. Unreal's large levels and rich textures take up a lot of virtual memory. We recommend having 150 megabytes of free hard disk space for running the game, and 300 megabytes or more of free hard disk space for the editor.

Video issues

Some older DirectDraw drivers do not support low resolution 16- and 32-bit color modes (for example, 320x240, 400x300, and 512x384). If you are playing Unreal using software rendering, this is unfortunate because the higher resolution modes like 640x400 and 640x480 run significantly slower than the low resolution modes. If the only full-screen options shown in Unreal are high-res, you might try obtaining a newer DirectDraw driver for your video card.

Some video drivers do not properly support DirectDraw full-screen in 16- or 32-bit color at any resolution. If you are unable to play Unreal full-screen, you should obtain a newer driver from your video board manufacturer. If you can't find a working DirectDraw driver, you can still play Unreal in a window, though the performance and immersiveness are not as good as playing full-screen.

On nearly all machines, Unreal runs optimally in 32-bit color mode (rather than 16-bit color). You can elect 32-bit color either from the main Unreal window. Alternatively, if you set your desktop resolution to 32-bit color, then Unreal defaults to 32-bit color.

To maintain a consistent frame rate despite a large number of translucent surfaces and/or explosion effects in software rendering, you might want to set "SlowVideoBuffering" to "True" in the "Advanced Options/Display" menu.

Known video issues:

- Unreal does not work on many 3dfx Voodoo Rush based 3d accelerators, due to driver conflicts which we haven't been able to resolve.
- On 3dfx under Windows NT, if you switch from the software renderer to 3dfx more than once per run of the game, you may receive an error message.

Sound issues

The sound playback rate defaults to 22 kHz, which is optimal for Pentium machines with MMX. If you have an older, non-MMX machine, you can get better performance (though lower sound quality) by setting the playback rate to 11 kHz in the "Advanced Options" menu. If you have a fast, shiny new Pentium II, you might try using 44 kHz for the ultimate in sound quality.

If using an Aureal A3D 3D sound accelerator board such as the Diamond Monster Sound 3D, you need to go into "Advanced Options/Audio" and turn on "Use3dHardware" to enable 3D sound card support. You need to upgrade to the latest version of Aureal's A3D drivers in order to get acceptable 3D sound performance. Using Unreal in conjunction with earlier versions of the drivers causes severe performance problems (major slowdowns on the order of 30-50% while playing sound).

If your computer is hooked up to a Dolby SurroundSound receiver, you should go into "Advanced Options" and turn on the "UseSurround" to take advantage of 360-degree Dolby sound panning, which rocks.

Known sound issues:

- A small number of computer configurations we've tested exhibit infrequent lockups when playing in full-screen DirectDraw (software rendered) mode using DirectSound for audio output. On most of these machines, going into "Advanced Options/Audio", and turning off the "UseDirectSound" option prevented the lockups. Playing in a window also prevented the lockups.
- Because of the way the OS works, many Windows NT machines experience significant latency in their sound effects, sometimes up to 1/4th second.
- If you have an Aureal A3D sound card with Unreal 3D sound enabled, you must disable the Aureal's "A3D Splash Screen" in the Aureal configuration utility. If this is not disabled, Unreal may be unable to go into full-screen mode because of the splash screen interfering.

Network play issues

The minimum speed connection for acceptable Internet play performance is a 28.8K modem connection to your Internet Service Provider.

When you become disconnected from an Unreal server, you are placed in a small holding level (which looks like a jail cell). From there, you can use the menus to reconnect or join a new server.

Known network play issues:

- When a new player enters a network game, clients may experience a 1/4-second pause while the mesh, skin, and other data is loaded for that player. This is by design.
- Unreal Internet play performance is highly dependent on the bandwidth of your connection, the latency (ping time), and the packet loss. The game is designed to be playable up to 300 msec ping times, 5% packet loss, and 28.8K connection speeds. Performance degrades heavily under worse latency, packet loss, and bandwidth connections.
- When switching between levels in a deathmatch game with a frag limit or time limit, or in a coop game at the end of a level, players are temporarily moved into the holding level (a small level that looks like a jail cell) for 5-10 seconds while the server switches levels and informs the clients of the level change.
- Coop savegames are not supported. Instead, the player who starts the coop game can specify the starting level.

Control issues

- **Known control issues:** Some PC keyboards can't recognize certain combinations of 3 or more simultaneously pressed keys.

ADVANCED OPTIONS

In Unreal, go into the "Options" menu then select "Advanced Options" to bring up the menu of advanced options. Some of the more useful options you can customize are:

Audio

AmbientFactor: Scaled ambient sound effects relative to regular sound effects. Can be 0.0 - 1.0, defaults to 0.6.

EffectsChannels: Number of simultaneously playing sound effects. Defaults to 16. Use a lower number to increase performance, at the expense of sound detail.

LowSoundQuality: Increases performance and reduces memory usage, by substituting lower-quality versions of sounds.

OutputRate: The sound playback rate. The higher the number, the better the sound quality and the slower the performance. 11025Hz is medium quality, 44100 is CD-quality.

Use3dHardware: Enables support for Aureal A3D sound cards.

UseDirectSound: Enables DirectSound support.

UseReverb: Enables and disables echos and reverb.

UseSurround: Enables Dolby SurroundSound(tm) support. Requires that your computer is hooked up to a Dolby SurroundSound(tm) receiver. Works with all sound cards.

Display

CaptureMouse: When enabled, causes Unreal to hide the mouse cursor when you are playing within a window.

CurvedSurfaces: Enables smoothing of monster meshes.

LowDetailTextures: Increases performance and reduces memory usage by substituting lower quality textures. Great for low-memory machines.

NoLighting: Turns off all lighting within the game. Looks ugly, but increases performance on low-end machines.

StartupFullscreen: Whether to start up with the game running in a window, or full-screen.

UseDirectDraw: Enables DirectDraw full-screen rendering support.

UseJoystick: Enables joystick support.

Drivers

GameRenderDevice: The driver for 3D rendering during gameplay. If you install new 3D hardware, go into Advanced Options and change this.

Game Types

For each major type of game supported by Unreal (deathmatch play, coop play, etc), you can set these options that affect gameplay. Very useful for internet UnrealServer administrators.

Joystick

InvertVertical: Inverts vertical joystick movement, for people who prefer flight simulator style controls.

ScaleRUV: Scales the sensitivity of the trackball or advanced joystick axes.

ScaleXYZ: Scales the regular joystick axis sensitivity.

Networking / TCP/IP Network Play

DefaultByteLimit: The default transmission rate. You should set this to reflect the speed of your Internet connection, in order to maximize performance:

- On 28.8K modem connections, use 2600.
- On 56K modem connections, use 3600.
- On LAN connections, use 25000.

Rendering

Here, there are rendering options for each 3D rendering driver installed on the system.

Coronas: Enables translucent coronas around lightsources.

DetailTextures: Enables special ultra-high resolution textures which add detail to complex surfaces when you get up close.

HighDetailActors: Enables rendering of high detail objects in the world. This should be turned on for fast machines, and turned off for slow machines.

ShinySurfaces: Enables shiny (reflective) surfaces.

VolumetricLighting: Enables space-filling, volumetric lighting and fog. Only visible on some 3D hardware drivers, and in the software renderer on MMX PC's.

PERFORMANCE TIPS

Our focus in creating Unreal has been to deliver a next-generation game that brings 3d gaming to a new level of realism. That is good. A side effect of this is that Unreal also runs slower than past 3d games on older or low-end PC's. That is bad...but it's an inevitable result of the large quantity of high-detail artwork; open, realistic, and high-detail environments; and high-detail animations in the game.

What follows are some tips on how to up Unreal's performance on machines where the game runs slowly.

Low Detail Settings

The "Advanced Options" menu contains many settings that enable you to trade off detail for performance. Here are the choices:

Display / Low Detail Textures: Trades memory for texture detail (resolution). When on, reduces memory usage by 5 megabytes on average. Recommended for slow PC's and PC's with low memory.

Audio / Low Sound Quality: Trades memory for sound quality. Turning this on reduces sounds to 8-bit, saving a significant amount of memory.

Audio / OutputRate: Trades speed for sound quality.

- 11025 Hz: Medium sound mixing quality; best for non-MMX machines.
- 22050 Hz: High sound mixing quality; the default.
- 44100 Hz: Ultra high sound quality.

Audio / EffectsChannels: Trades speed for sound realism. The default is a highly realistic 16 channels of sound. On slower machines, you may want to change this number to 8 or 12.

Memory

Unreal's performance is highly dependent on the amount of RAM you have in your machine, and the amount of memory that is available. Machines with less memory will access their hard disk more frequently to load data, which causes sporadic pauses in gameplay. Thus, if you have a 32 megabyte (or less) machine, you should make sure that you don't have other unnecessary programs loaded in memory when playing Unreal.

How Unreal will perform under different RAM conditions:

- Less than 16 megabytes: Unplayable.
- 16 megabytes: Playable, but very frequent swapping to the hard disk. We highly recommend turning on "Low Detail Textures" and "Low Quality Sound" to reduce memory usage.
- 32 megabytes: Some swapping.
- 64 megabytes: Great, with perhaps a teeny bit of swapping.
- 128 megabytes: Oh Yes!

CPU Speed

Unreal is also very sensitive to CPU speed, memory bandwidth, and cache performance. Thus, it runs far better on leading-edge processors such as Pentium II's than it does on older ones such as non-MMX Pentiums.

How Unreal will perform on different classes of machines:

- Non-MMX P166 class machines: Slow rendering; large frame rate variations. We recommend playing in 320x200 resolution if available. We recommend setting the sound playback to 11025 Hz.
- P200 MMX: Good rendering speed; some frame rate variations. We recommend running low resolutions ike 320x240 or 400x300. We recommend keeping the sound playback at 22050 Hz.
- Pentium II: Very nice rendering speed; consistent frame rate. Software rendering runs smooth in 512x384, 32-bit color resolution. You might try 44 kHz audio for best sound quality.

Considering upgrading?

For people considering upgrading their machines, here are some tips based on our experience running Unreal on a variety of machines:

1. The biggest performance gain in Unreal comes from having a Pentium II class processor. Pentium II's have dramatically improved cache performance, memory performance, and floating-point performance compared to earlier Pentiums, and that all translates to faster gameplay. The performance improvements in Pentium II's are especially accentuated in Unreal, which contains way more content (textures, sounds, animations, level geometry) than other 3D action games.
2. The next upgrade that tends to improve Unreal performance dramatically is a 3dfx Voodoo or Voodoo2 class 3D accelerator. Especially in conjunction with a Pentium II processor, these accelerators rock!
3. Finally, lots of RAM helps. With memory prices continually falling, it's now reasonably affordable to upgrade to 64 or 128 megabytes of memory.

Requirements

Minimum system requirement:

- * 166 MHz Pentium class computer.
- * 32 megabytes of RAM.
- * 2 megabyte video card.

Typical system:

- * 233 MHz Pentium MMX or Pentium II.
- * 32 or 64 megabytes of RAM.
- * 3dfx Voodoo class 3d accelerator.

Awesome system:

- * Pentium II 266 or faster.
- * 64 or 128 megabytes of RAM.
- * 3dfx Voodoo or Voodoo2 class 3D accelerator.

CONTROLS

Keyboard

Up Arrow:	Move forward
Down Arrow:	Move backward
Comma:	Strafe left
Full Stop:	Strafe right
Mouse Movement:	Rotate view
Control, Left Mouse Button:	Primary fire
Alt, Right Mouse Button:	Alternate fire
Space:	Jump
Enter:	Activate selected inventory item
Shift:	Toggle running
Pause:	Pause the game
Z:	Strafe (cause the arrow keys to strafe)
◀:	Turn left
▶:	Turn right
1,2,3,4,5,6,7,8,9,0:	Change weapon
/:	Switch to next available weapon
-, +:	Resize game window
[,]:	Select inventory item
F2:	Activate/Deactivate Translator
F5:	Change HUD

Network Play Keys

F:	Feign Death
::	Lob/Throw current weapon
F4:	Display scoreboard
T:	Type a message
L:	Wave to other players.
J:	Taunt other players.
K:	Victory taunt.

Mouse

Although you can rely solely on your keyboard to move around in and interact in Unreal's 3D universe, using both the keyboard and mouse simultaneously gives you much more fluid and responsive control.

When you use the mouse to control your rotational movement and aiming you gain a degree of precision and speed that players using keyboard-only controls can't touch. The keyboard is best used for easy lateral and forward/backward movement, and for jumping.

To master the default controls in Unreal, keep your left hand on the keyboard, using the arrow keys for movement, the 0-9 keys for weapon selection, and the space bar for jumping. Your right hand operates the mouse, controlling rotation, aiming, and firing. Of course, you can customize these controls to suit your preferences via the Options Menu.

Joystick

You can enable joystick support in Unreal through the Options menu (it's off by default). You can use a standard joystick for movement and firing. In addition, Unreal has built-in support for the Panther XL joystick that supports dual joystick and trackball play.

Standard joystick controls:

- Move/Rotate: Joystick handle
- Fire: Button 1
- AltFire: Button 2
- Jump: Button 3 (on 4-button joysticks)
- Duck: Button 4 (on 4-button joysticks)

Additional panther XL controls:

- Strafe/Look: Trackball

Customizing the controls

If you don't like the default controls, you can change them by going into the "Options" menu and selecting "Customize Controls".

INTERNET AND LAN GAMES

Go into the "Multiplayer" menu to start or join a network game.

Unreal supports both LAN and Internet play with the standard TCP/IP protocol. If you have an Internet connection, you should be ready to go!

DEDICATED NETWORK SERVERS

Explanation

For optimal network play performance, you can launch a dedicated copy of the Unreal server on a computer. This improves performance compared to using a non-dedicated server but, of course, it ties up a PC.

Launching

You can launch a dedicated server by going through the regular Unreal "Start Game" menu, setting the appropriate options, then choosing "Launch Dedicated Servers". This is what you'll want to do for quick LAN games where you have an extra machine sitting around that can act as a dedicated server.

Alternatively, you can launch a dedicated server from the command line by running Unreal.exe directly which usually resides in the c:\Unreal\System directory, or the System subdirectory of whatever other directory you installed the game in). For example, to launch the level "DmFith.unr", run: Unreal.exe DmFith.unr -server

Multiple Servers Per Machine

Each copy of the Unreal dedicated server can serve one and only one level at a time.

However, you can run multiple level servers on one machine. To do this, you must give each server a unique TCP/IP port number. Unreal's default port number is 7777. To specify a port, use the following kind of command line: Unreal.exe DmFith.unr port=7778 -server

Some Windows NT servers may have more than one network card installed, and thus more than one IP address. If this is the case, you need to specify the IP address for Unreal to play on using the multihome=www.xxx.yyy.zzz parameter such as: Unreal.exe DmFith.unr multihome=204.157.115.34

General performance guidelines

We find that a 200 MHz Pentium Pro can usually handle about 16 players with decent performance. The performance varies with level complexity and other machine speed factors, so your mileage may differ. Note that there is no absolute maximum player limit in Unreal; performance simply degrades as the number of players grows huge.

If you're running multiple levels simultaneously, Windows NT outperforms Windows 95 because of its superior multitasking and TCP/IP processing capabilities.

For best performance, we recommend having 32 megabytes of memory per running level. For example, for running 4 simultaneous levels, 128 megabytes is ideal.

The Unreal server uses up at least 28.8Kbits per second of outgoing bandwidth per player (on Internet), so if you run the server on a machine connected by a 28.8K modem, you'll only be able to support one client with decent performance. Dedicated servers that support many players generally require the outgoing bandwidth of a T1 line or better.

Master Server Uplink

If you're running a public Internet server and you want it to be listed in Epic's public Unreal server listing, go into "Advanced Options", "Networking", and "Master Server Uplink" to turn on the "DoUplink" option. This will cause your Unreal server to contact Epic's master Unreal server at master.unreal.com every 30 seconds and advertise its IP address and player list.

Note: This option is OFF by default. If you enable it, this option causes your IP address, server name, server options, and player lists to be advertised to the world.

THE EDITOR (QUICK INTRO BY CLIFF BLESZINSKI)

Things To Keep In Mind

Don't be afraid to try things in the editor. If worse comes to worse, you can always re-install anything that you break. Feel free to add in creatures, screw with their properties, build crazily detailed scenes, make funky lighting schemes, etc. The best way to learn is by doing.

To run the editor, run UNREALED.EXE in your Unreal System directory.

Now, the editor is made up of three main viewing windows. On the left side of the screen we have your main toolbar, this has buttons for most of your functions. In the middle we have all of your camera views, including overhead, side, and first person. On the right side

we have your class/texture browser. From that section you'll pick textures, enemies to add, music, and sound effects. Remember the terms Camera Views, Toolbar, and Browser, as I will be referring to these quite frequently.

Let's try loading some textures to start. On the right side texture browser, make sure it says BROWSE-TEXTURES in the drop down boxes. Now, at the bottom, hit LOAD and you can load UTX texture packages. Unreal stores 8 bit PCX files in its UTX packages. Don't modify the main ones, or various levels will not run.

Now that we have some textures loaded, we're going to discuss how you edit in Unreal.

Unreal is based on Constructive Solid Geometry. Try to imagine that the whole world is already filled up, and you have to carve your level out of it. You can overlap shapes, you can intersect shapes, and you can add shapes into holes/rooms that you already cut out of the world.

On the third column of the toolbar, approximately halfway down, you'll see a cube, cone, sphere, and a couple of staircases. These are your primitives. For building your basic shapes in your world, they're invaluable. You can right click on any of these to edit their properties, this is the best way to scale and resize them. Click on the cube, and a red cube outline will appear in your windows. This is your building brush. You only have one building brush in the world at a time, it is designated as a red outline. Now, this brush isn't actually a piece of world geometry yet, it is more like a rubber stamp that you use to punch holes and add into the world with. With any brush you can add it in, or cut out with it.

Note: Unreal defaults to a grid of 16 units. You can change this by right clicking on the 2d camera views and selecting Grid->x units.

Now that you have a cube that you can work with in the world, you're ready to start adding and subtracting. First, pick a texture in your texture browser. Now, on the toolbar, click the second button down on the right column. This is your subtract button, you can subtract by hitting this or by pressing CTRL-S. You should see an empty room now in your first person camera view!

Note: To move around in your 2d camera views, left click and drag to change your view. Hold both mouse buttons and move the mouse to zoom in and out. In the 3d views, to move forward and back, left click and move the mouse to move. Hold the right button to look around in the 3d view. Hold both buttons and move the mouse to strafe and move up and down.

You can move your building brush around by holding CONTROL and left mouse clicking in the 2d camera views. To rotate your building brush, hold CONTROL and right click and drag the mouse in the 2d views. If you want to scale your brush on the fly, click on the fourth button down on the left column of the toolbar, you'll be in SCALE TO GRID mode. (To get out of this mode, or any other modes, click the top left eyeball on the toolbar.) To scale your brush on the grid, move to one of the 2d views and hold control while left clicking and dragging. If you want to reset your brush's rotation, location, or scaling, right click on your builder brush and click RESET- rotation, scaling, or location.

Rebuilding

To see your updated geometry, you'll need to do a quick rebuild. You can go OPTIONS- REBUILD, or right click in the 2d views and go to REBUILDER. A window will pop up, this is your REBUILDER. Right now there are 3 parts to this, you can see each part to rebuilding on the tabs in the rebuilder window. They are GEOMETRY, BSP, and LIGHTING. For now, you can disregard BSP. When you first bring up the rebuilder window, it will have AUTO REBUILD BSP checked ON, turn this off. You can now rebuild GEOMETRY and LIGHTING separately, without doing anything to the BSP. You can add and subtract brushes all you want, but if you want to see your level accurately you'll have to rebuild geometry. This goes for lighting as well, you can add and edit lights all you want, but to see things properly in the editor you'll have to rebuild lighting.

You can select any surface in the world by left clicking on it. You can select multiple surfaces by holding Control and left clicking on the desired surfaces. To edit a surface's properties, right click on it. To edit multiple surfaces, select them while holding Control and continue to hold control when you right click on one of the selected surfaces. You can go to surface properties on any of them, where you can adjust their scale, you can align textures on surfaces, pan/flip/rotate the textures, etc.

Try experimenting with the primitive shapes and seeing what you can come up with. Once you build a room or two, it is time to add lighting.

Lighting

To add a light in Unreal, you can right click on any location and click ADD LIGHT HERE on the menu that pops up. Or, you can hold the L key on the keyboard and left click anywhere to add a light. You'll see a little torch icon that'll appear where you clicked.

This represents where your light source is. You can right click on any light to edit its properties. Under Lighting and LightColor, you can edit the light's brightness, radius, hue, and saturation. There are also lighting effects you can experiment with, such as flicker, pulse, and disco.

Note: To recalculate and see your proper lighting, you'll need to do a quick rebuild of lighting. See Rebuilding above.

You can just left click to select a light, and you can then hold Control and left mouse click to select multiple lights. Now, hold Control and left mouse click and drag in the 2d views to move the lights around in the 2d views. To deselect lights, hold Control and left click again on various lights. You can deselect anything by clicking the big 0 on the toolbar.

Adding Goodies

You can add monsters, items, and decorations just as easily as you add lights in the Unreal Engine. First, you need to find them: Go to the texture browser and where it says BROWSE: TEXTURES, and click on the drop down box and select CLASSES. Your browser is now viewing actors. (Actors include monsters, items, keypoints in the level, and decorations.)

To add in a monster, expand the PAWN menu. Then, expand SCRIPTED PAWN menu. All of your monsters reside in there. To add them in, select a monster type, and then right click and hit ADD (Monster type) HERE. Or, you can just hold A and left click anywhere to add that actor.

To add an item, expand INVENTORY. In that area, you have all your pickups, items, and weapons.

Adding decorations is just as easy. Just expand DECORATIONS, select an actor, and add away.

You can edit the properties for any monster, item, or decoration by right clicking on it and going to its properties window.

To play your level, you'll need to let the game know where to put the player when he enters the level. This is done by adding in a PLAYERSTART actor, it is located underneath NAVIGATION POINT-> Playerstart. When you run the level you'll start here.

When you're running the game, you can hit TAB or Tilde (`) to enter console commands. Typing in STAT FPS will give you a frame rate/polygon count display. Unreal's target polygon counts are:

- 200: No combat going on, little to no actors or monsters out there.
- 130 or less: Combat with one monster.
- 100: Combat with 2-3 monsters.
- 20-50: Combat with 4 monsters or less.

Intersecting and De-intersecting

One of the most powerful features of the editor, Intersecting and De-Intersecting allow you to capture any existing shape or empty space in your level for the purpose of creating an all new building brush. Try this: Build a shape in your level, such as a chair. Then, build a cube that encapsulates that chair. In the top menu bar, go BRUSH-> INTERSECT. This takes all the solids that are within the boundaries of your cube and captures it as its own brush. You can now move that new builder brush around and add/subtract all you want You can intersect/de-intersect with any shape, at any time!

Creating Moving Brushes

A Moving Brush, or Mover, is a piece of the world geometry that moves in predefined keyframes of animation. A typical mover is a door, a drawbridge, or a collapsing plank.

The best way to create your movers:

1. Make a huge building box somewhere off to the side of your level.
2. Build your lift, door, or whatever in that building box.
3. INTERSECT what you built.
4. Move your new red builder brush into your level and place it where you'd want your mover.
5. Hit the sixth button down on the right side of the toolbar, that looks like a cube with gray arcs coming off of it.

6. This added in your mover. You need to set its mover keyframes now, right click on it and say KEYFRAME-> 1. Move your mover to keyframe 1, and right click on it again and go KEYFRAME-> 0 to reset it. (NOTE: Your mover will be a purple brush, it will appear beneath your red builder brush. Be sure to move your red builder brush out of the way.)

7. Your mover is in place. In the movers' properties, under OBJECT, you can specify what makes this mover move. It defaults to BUMP OPEN TIMED, which means the mover will open when bumped into by the player. There are other types in here, mess around with them!

The multiple mover activate types are:

- Stand Open Timed: The mover will activate and go from keyframe zero to one when the player stands on it. Set lifts to this activate type.
- Bump Button: Um, I don't use this much, don't bother with it.
- Bump Open Timed: When the player bumps this moving brush (touches it) it will remain open for the time specified in MOVER (stay open time.) NOTE: The MOVE TIME is the time it takes the mover to move from keyframe to keyframe.

You can also activate a mover by a trigger. A trigger is an invisible actor that you can add into the world to activate most anything, in this instance, a moving brush. Triggers are in your class browser under TRIGGERS-TRIGGER. Add one in. A trigger, under EVENTS of its properties, has two areas you need to know about: EVENT and TAG. The EVENT is the name of what that this trigger will cause when it is touched/shot/activated. This corresponds to the TAG of the action you want to occur. For instance, a trigger that has the EVENT of Fred will trigger the moving brush that has the TAG of Fred. You can use his system to daisy chain events, you can have the Fred mover's event be George, and have another mover hat's tag is George. That mover will trigger when the first one is done (provided it is set to Trigger Open Timed which leads us to the next mover activate types.)

- Trigger Pound: When you are touching the trigger who's EVENT matches this mover's TAG, the trigger will pound between its keyframes. Useful for squishing timing puzzles, or machinery.
- Trigger Control: When the player is touching the trigger for this mover, the mover will start to open and remain open as long as the player is within the vicinity of the trigger. Once he leaves the trigger, the door will proceed to close. Useful for doors.
- Trigger Toggle: When the trigger is touched, it will open the mover it is tagged to. When the trigger is touched again, the mover will close. It toggles the state of the mover from open to closed.
- Trigger Open Timed: When the trigger is touched, the mover will open, stay open for said time, and close.

You can also have one moving brush (lovingly often referred to as a Mover) trigger another. Build a button whose Object-State is BUMP OPEN TIMED and match its event to a door mover's tag. Set the door mover's Object-State to TRIGGER OPEN TIMED, and the door will open when the button is touched. (provided you specified keyframes, etc)

A helpful property in the movers' properties is BTRIGGERONCEONLY, when this is set to true the mover will open and never close again. Useful for one time doors/puzzles/traps.

Let there be Fire

As you've probably already noticed, most of the torches and fire effects created in Unreal are simply single polygon sheet brushes that intersect in an asterisk pattern to create the illusion of volume.

To create your own torch flame, follow these steps:

1. Load a fire UTX (one of the pre-made ones are easier for now, such as the GREATFIRE series) and elect your fire texture.
2. You'll need to build your sheet builder brush in the world. On the toolbar, the bottom right button looks like a gray diamond shape) will build your sheet. Right click on the button and edit the size/orientation. You'll want X or Y axis, try the default 128x128 size.

3. Now, click on the FIFTH button down on the toolbar (looks like a regular green square.) A window box will pop up, this is your ADD A SPECIAL BRUSH dialogue box. Check 2 SIDED (so the sheet will be visible from front and back,) TRANSPARENT (so you'll be able to see through the fire,) and NON SOLID (non solid brushes do not cut up your BSP and add polygons to your world.)

4. Now, in that dialogue box, press ADD SPECIAL. Your sheet will appear in the world.

5. In the overhead view, rotate your builder sheet brush 45 degrees or so, and proceed to make a crossing pattern, like an asterisk.

6. Right Click on your fire textures and highlight UNLIT.

7. You're ready to go! Add a base shape below your fire, add a light source, rebuild, and you've got a torch in your world.

Cheats

Hit TAB and type in to execute.

- Allammo, gives you 999 ammo for all your guns.
- Fly, lets you fly around.
- Walk, use this to stop flying or GHOSTING.
- Ghost. Noclip through walls.
- PlayersOnly, freezes time. Press again to resume time passage.
- GOD: God mode.
- OPEN MAPNAME: Jump to any map, just enter the name like OPEN DIG.
- SUMMON itemname: Adds whatever you want to the world.

Some stuff you can add:

- SUMMON EIGHTBALL
- SUMMON FLAKCANNON
- SUMMON NALI
- SUMMON SKAARJWARRIOR

- BEHINDVIEW 1: Puts you in Tomb Raider style view. BEHINDVIEW 0 resets this. (be sure to show off the shieldbelt environment mapping this way, and the cool animations, and the cool swimming...)
- FLUSH: If you start getting weird garbage graphics on wall textures or creatures, type this. It gets rid of this.

For More Editor Info

Visit the Unreal home page on the Web, <http://www.unreal.com/>, for the online documentation for UnrealEd.

UnrealEd is a memory hog -- it consumes over 100 megabytes of memory (real or virtual) when editing a typical level, and up to 200 megabytes when rebuilding the most detailed levels.

UnrealScript

UnrealEd contains a full compiler and development environment for UnrealScript, Unreal's built-in programming language. UnrealEd enables you to view, edit, compile, and save scripts.

UnrealScript is an object-oriented language similar to Java. It is by far the most thorough and advanced programming language ever built into a game.

To view scripts, go into UnrealEd's class browser (in the panel on the right hand side of the screen, select the "Classes" tab). Here, you see Unreal's class hierarchy. At the base of the hierarchy is Actor, the class which represents all objects which can exist, move around, and interact within a level. There are many others, such as:

- Inventory, which represents all objects players can carry.
- Weapon, all weapons that players can use.
- Pawn, all monsters and players.
- ScriptedPawn, all advanced AI-controlled monsters.
- PlayerPawn, all players.
- Projectile, all weapon projectiles.
- Decoration, purely decorative objects.

Full documentation on UnrealScript is available on the Unreal web site at <http://www.unreal.com/>.

CREDITS

Development Team

Game Design: James Schmalz & Cliff Bleszinski. Level Designers: Cliff Bleszinski, T Elliot Cannon, Cedric Fiorentino, Pancho Eekels, Jeremy War, Shane Caudle. Animator: Dave Carter. Artists: James Schmalz, Mike Leathem, Artur Bialas. Programmers: Tim Sweeney, Steven Polge, Erik de Neve, Carlo Volgelsang, James Schmalz, Nick Michon.

Musicians: Alexander Brandon, Michiel van de Bos. Sound Effects: Dave Ewing.

Epic Biz: Jay Wilbur, Mark Rein, Nigel Kent, and Craig Lafferty.

For GT Interactive

Producer: Jason Schreiber Executive Producer: Greg Williams Lead Tester/Associate Producer: Joel

Maximillion Breton Product Manager: Ken Gold Assistant Product Manager: Phil Tucker Public Relations

Manager: Alan Lewis Director of Creative Services: Leslie Mills

Creative Director: Vic Merrit Artists: Michael Marrs, Jill Pomper, Lesley Zinn, and Jen Scheerer Production

Coordinator: Liz Fierro Box Design: Vic Merrit and Leslie Mills

Special Thanks

Chad Faragher, Nick Oddson, Chris Hargett, DJ Carroll, Diane Schmalz, Shannon Newans, Evelyn Eekels,

Lani Minella, Gina Hedges, Ryan Schwartz, Mark Visser, Richard Young,

Mike Forge, Eric Reuter, and the guys at UnrealNation and Unreal.org.

Testing

Lead Tester: Mike Barker, Second: Jim Tricario, Second: Dan McJilton

Testers: Mike Barker, Jim Tricario, Dan McJilton, Dave Munro, Andre Cerny, Cormac Russell, Jesse

Smith, Clint McCaul, Fran Katsimpiris, Corey Allen, Ed Piper, Barry Gilchrist, Adam Coleman, Chris Carr,

Chris McGuirk, Randy Denmyer, Kevin Keith, Thomas Watkins, Dave Afdahl, Andy Mazurek, Matt

Kutrik, Troy Kupich, Jake Grimshaw, Mark Leary, Matt Miller, Ian Giffen, Justin Dull, Calvin Grove,

Ruben Brown, Mike Prendergast, Geoff Gessner, Steven Rhodes, Rocco Rinaldi, Jim Biltz.

THE UNREAL MASTER PLAN

The initial release of Unreal consists of the game and an unsupported beta version of the editor. The game is complete and contains all of the planned game features and hardware support for the initial release. The editor is fully usable, and it's an exact replica of the editor the Unreal team has used to build our game, but it's not as stable and user-friendly as we'd expect a full product to be, and the documentation hasn't been written yet, thus it is labeled as a beta.

Updates Coming

After release, our current plans call for making several follow-on updates available freely to Unreal players on the Web:

1. OpenGL and Direct3D support, enabling 3d hardware acceleration on many cool non 3dfx cards such as the Intel i740, RIVA 128, Rendition cards; and many upcoming next-generation 3d cards.
2. UnrealServer performance and stability enhancements. While we've tested network play thoroughly internally before release, this is our first big Internet-playable game, and we expect to learn a lot based on the feedback of thousands of players, and make improvements. Additionally, hackers may find ways to crash or degrade performance on UnrealServers, and we expect to release updates as problems are discovered.

Additional Products

Stay tuned for the following upcoming retail products:

1. Unreal Editor full version (retail product), a much-improved, more user friendly, more stable, and fully documented Unreal editor available in retail stores and direct from Epic. Watch our web site (<http://www.unreal.com/>) for information about this.
2. The official Unreal level pack, coming to retail stores from GT Interactive.
3. Hint books, strategy guides, books, and any other Unreal related paraphernalia those crazy GT Interactive marketing people dream up!

NOTES

Get unstuck with the



HINTS AND TIPS

0891 669910

THE UNREAL HINTS & TIPS HELPLINE IS CHARGED AT 50p A
MINUTE AT ALL TIMES.

PLEASE MAKE SURE YOU HAVE PERMISSION FROM THE PERSON
WHO IS PAYING THE BILL, BEFORE CALLING..