

Unreal Technology Features

Tim Sweeney

Epic MegaGames[?], Inc.

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

A Complete Solution

The biggest strength of the Unreal technology, relative to third-party and homegrown tools, is that it provides a complete solution that has stood the tough test of real-world **game development**. While there are certainly some quality third-party game components on the market, such as QSound, Motivate!, and Caviar, they all provide only a small part of the overall 3D game-**development equation**. The saying "God is in the details" is very applicable to game engine development: a cool rendering algorithm can only be exploited if it integrates seamlessly into an engine's collision, visibility, networking, AI, and file management subsystems. All of Unreal's components have been designed together; they provide a consistent programming interface; and they fully interoperate.

As a technology developed in conjunction with a leading-edge, high-detail game, the tradeoffs that Unreal makes on detail versus realism are fundamentally sound and tailored to real-world constraints. For example, some people may criticise Unreal's cylindrical actor collision system as being low-tech compared to an IK-based system. However, this decision was made with an accute awareness of all the ramifications it had on the overall product: cyllindrical collision makes AI, player control programming, and network predication much simpler, and it keeps performance up. A collision system designed in isolation could easily be more advanced than Unreal's, and it would probably cause countless headaches when making it work with the rest of the codebase and making it run at a decent frame rate on 1998-1999 PC's.

Editing Tools

- UnrealEd is a realtime level design tool based on constructive solid geometry, optimized for building realtime 3D environments.
- UnrealEd is fully integrated into the Unreal engine, so the camera views are what-you-see-is-what-you-get: all lighting, texture placement, and geometry operations take effect immediately, providing extremely fast feedback while building levels.
- UnrealEd "brushes," the basic units of CSG, can be unlimited in size: no limitations on polygon counts or convexity.
- Brushes can be generated within the editor using many built-in tools such as box, cylinder, sphere, staircase, spiral staircase, and cone.
- Brushes can be imported from .DXF or .ASC files.
- Visual editing of properties of actors and objects.
- In UnrealEd[?], you are one mouse click away from launching your level for play testing.
- "Texture browser" tool for easily selection and replacement of textures.
- "Surface properties" tool for easily modifying attributes of surfaces such as alignment, scaling, panning, and special effects.
- "Actor properties" tool for visual editing of actors, similar to Visual Basic.

Mesh animation

- All mesh animations (creatures, weapons, etc) can be imported as frames from .DXF files output from state-of-the-art tools like 3D Studio Max, Alias PowerAnimator[?], SoftImage[?], or motion-capture tools.
- Mesh animations support interpolation enabling perfectly fluid animation at high frame rates. With interpolation, an animation recorded at 15 fps can be played back at 40 fps perfectly smoothly.
- Support for "tweening" between disjoint animation sequences, enabling monster and player animation to smoothly flow between actions.

Rich, organic surface support

- **Texture mapping** for applying base detail to surfaces.
- **Multicolored light and shadow mapping** for applying complex, dynamic, multicolored lighting to surfaces with shadowing and numerous special effects, including water refraction caustics.
- **Fog mapping** for applying atmospheric and volumetric lighting and fog effects to three-dimensional volumes of space.
- **Detail texture mapping** to add extremely close-up details to surfaces, such as fine wood grain, pock marks on brick surfaces, or scratches to metal surfaces.
- **Macro texture mapping** for adding very large scale detail to surfaces, for example, adding erosion patterns to an outdoor landscape, or applying a hand-painted facade to a large building.
- **Parametric, fractal, and animated texture mapping** support for creating organic animating materials with endless variety, such as fire, realistically wavy water, scum, ice, puddles, and so on.
- **Translucency, modulation, and masking** texture effects.

UDN

Search public documentation:

[Search](#)

TechnologyFeatures

Licensees can [log in](#).

[Red](#) links require licensee log in.

Interested in the Unreal Engine?
Visit the [Unreal Technology](#) site.

Looking for jobs and company
info?

Check out the [Epic games](#) site.

Questions about support via UDN?

Contact the [UDN Staff](#)

- **Sky portalization** enables level designers to build complex "sky volumes" containing realistic sky, stars, moons, suns, geometry, etc. and see out of their level into the sky box, as if the sky box is infinite in size.
- Up to **12 levels of mipmapping**.
- Supports **texture animation sequences** with variable playback rate and automatic synchronization to one-shot special effects like explosions.

State of the art mesh support

- Curved-surface rendering support, with an adaptive level-of-detail subdivision surface rendering algorithm, eliminating **polygonization**. This makes **organic objects**, such as the alien cows in Unreal, exceptionally smooth and realistic.
- Supports environment mapping and chrome mapping, for reflective objects like those in *Terminator 2*.
- Parametric mesh deformation enables meshes to be made fatter or thinner to add variety to organic objects.
- Character models may be "skinned" with up to eight independent texture maps of sizes up to 256x256, enabling extremely high-detail texturing.
- Variable translucency, which enables transparent objects, like explosions and special effects, to fade out smoothly.
- Modulation blending, enabling smooth, colored smoke and darkening effects.

Artificial Intelligence

- Built-in AI support for advanced creatures and bots.
- Bots know how to use all player movement options, weapons, and inventory.
- Bots know how to use switches, and navigate past platforms, and doors.
- Physics based AI makes complex actor movement straightforward.
- Animation system ties into AI via notification system, with animation playing, looping, and tweening.
- Path node based AI navigation system, with support for complex route evaluation, alarm points, and stasis states.
- Object orientation makes AI extremely expandable and customizable.
- AI properties are exposed to level editing tool for level designers to tweak.

Digital Sound System

- Supports 3D positioning via distance attenuation, stereo panning, doppler shifting.
- Supports software Dolby Surround sound encoding for full 360-degree panning among the center, rear, left, and right channels (requires a Dolby decoder).
- Supports rich, ambient environmental sounds which can be fully controlled from UnrealScript[?].
- Advanced software mixer supports up to 64 stereo channels of music and sound, and mixes to a 32-bit buffer internally for maximum sound quality.
- Supports 8- and 16-bit samples using any playback rate from 4 kHz to 44 kHz.
- Dynamic sample interpolation enables high-quality pitch shifting without distortion.
- **Optimized to take advantage of Intel's MMX™ technology, with fallback to support regular Pentium® processors.**
- Supports DSP-style reverb and echo postprocessing on the sound effect stream.
- Aural raytracing realistically simulates the echo and reverb characteristics of complex **environments**. When enabled, UnrealEd[?] pre-computes a reverb kernel for each designer-specified zone in the level.
- Supports Microsoft DirectSound[?] and legacy WinMM[?] sound output for backwards compatibility.
- Plug-in interface makes it possible to integrate other sound systems into the Unreal technology.
- Supports Aureal A3D 3D sound positioning, and the upcoming Creative Labs 3D sound cards.

Digital Music System

- Supports both CD audio and **realtime digitally mixed music**. Game designers can mix and match soundtracks in both formats to take maximum advantage of the speed and quality of CD music and the dynamic music and Internet portability of digitally mixed music.
- Smooth and instantaneous dynamic music changes for fading between songs, fading between tracks within a song, fading to silence, and instantly switching songs.
- Realtime mixing of musical instruments and sound effects.
- Supports volume and panning envelopes for professional composing.
- Supports up to eight octaves per sample.
- Supports all widely used module file formats, including .mod, .s3m, .xm, and .it.

Lighting

- Multicolored lighting with true colored intermixing of fuzzy shadows.
- Supports raytraced and enveloped lighting.
- Radial, cylindrical, spotlight, searchlight, ambient, spherical, shell, and 20+ special effect lights.
- Caustic effects such as "fire waver", "watery shimmer", and the like can be applied to lights.
- Realtime editing of lights in UnrealEd[?].
- Multicolored lighting supported both on 3D hardware and in all software rendering modes (with and without MMX technology). This gives designers a far more consistent look and feel on all target hardware than is achievable with engines which only support monochromatic lighting in software.
- Supports lens flares and coronas.

Rendering

- Software rendered 16- and 32-bit color with special optimizations for MMX technology and Pentium II processors.
- Hardware rendered 16-bit color (24-bit coming when supported by hardware).

Technical programming

- Programming APIs in C++ and UnrealScript?
- Complete four-way calling interface between C++ and UnrealScript?, enabling development of complete projects in either language.
- Based on rich, extensible, object-oriented class hierarchy.
- Highly modularized and replaceable, with all Unreal I game code cleanly segregated from all general engine code.
- C++ interface based on object model similar in style to MFC, the applications framework used by nearly all non-game Windows applications.
 - Supports dynamic loading of DLL's and scripts on demand, for modularity and efficient memory usage.
 - Robust debugging environment, with VC++ debugger support; flexible assertion system; and try-catch call stack display for tracking down errors in the field.
- UnrealScript interface based on object model similar to Java.
 - Garbage collected.
 - One hundred percent portable: UnrealScript? code runs on all supported platforms without recompiling.
 - "Everything is an object."
 - Safe client-side "sandbox" execution model.
 - Fully featured set of datatypes and operators.
 - Native language support for intuitively constructing hierarchical finite state machines (HSFM's).
- Performance optimization tools.
 - UnrealScript profile.
 - Comprehensive game, engine, and networking performance counters.

Game programming

- Game programming via C++ or UnrealScript?
- Object-oriented class hierarchy that maps naturally onto game concepts (players, monsters, inventory, triggers, etc).
- UnrealScript native language support for states and state scoping of functions, time-based execution and latent functions, and networking. This includes replication, priority management, reliability, and remote procedure calls (RPC), enabling developers to easily optimize their content for Internet play.
- Advanced AI, pathfinding, and navigation system.
- Powerful, fully customizable physics system.

Networking

- Unreal includes the most advanced game networking capabilities of any game to date.
- Supports both small LAN games and large-scale, server-based Internet games.
- Gamers can travel between UnrealServers? on the Internet much like browsing Web pages, with teleporters providing a link between servers.
- Automatic downloading of new content (levels, textures, sound, models, scripts) while browsing the Unreal universe.
- Java-style client-side scripting capabilities enable high performance even over slow 28.8K Internet connections.

Portability

- Modularization of Unreal's platform-specific code makes the engine fairly portable.
- Already up and running on MacOS? with 3D hardware support and Mac<->PC networking.
- Portable to high-end consoles systems with 16+ megs of RAM. Next-generation consoles are appropriate but current systems like the N64 and Playstation are not feasible.

Scalability

- Built on a highly modular, object-oriented foundation, the Unreal technology is designed to evolve through many generations of technological advances.
- There are very few hardcoded limits in Unreal. Scenes and levels of arbitrary complexity are supported; the only factor that limits complexity is virtual memory and speed.
- There are no limits on the sizes (dimensions) of polygons, enabling the creation of **large outdoor spaces**. Such areas perform quite well as long as visible polygon counts are kept within reason. This is a significant advantage over "surface cache" based rendering, which artificially restricts the size of surfaces.

Realtime algorithmic texture engine

- Wavy textures for producing on-the-fly planar displacement mapped textures with waves such as ponds, oceans, lava, and puddles.
- Algorithmic field-based realistic fire, explosion, particle, smoke, and steam effects.
- Animated ball and line lightning.
- All effects are fully customizable with a gigantic number of adjustable parameters and drawing modes.

- All effects are algorithmically generated in realtime, enabling seamless animation that never repeats.

Dynamic scene graph technology

Dynamic scene graph (DSG) technology is a natural extension of portal technology. A dynamic scene graph consists of a root node, corresponding to the player's immediate surroundings, and a hierarchical tree of child nodes. Each DSG node has a coordinate system transformation, clipping volume, and other rendering state associated with it. Unreal's rendering engine constructs a dynamic scene graph on-the-fly as each frame is rendered. New scene graph nodes are generated as the viewer's visibility propagates outward through portals. Scene graph nodes are the basic building blocks of many realistic effects and special effects, such as:

- Mirror surfaces.
- Semireflective materials, such as marble surfaces which partially reflect light.
- Non-euclidean, redirectable "warp" portal effects for seeing through teleporters.
- Seeing through windows into an infinite sky zone in which a sky, planets, mountains, and other objects are constructed.
- Skies and backgrounds with independent coordinate systems for independent translation and rotation.

The strength of DSG rendering is that all of the node effects are fully interoperable. For example, once a mirror DSG node is defined, mirrors are automatically supported recursively (for hall-of-mirror type effects); mirrors work with skies; mirrors work with warp portals; warp portals work with skies, etc.

Unreal's dynamic scene graph technology is a basic building block which will be extended in future versions of the technology to incorporate such next-generation effects as seeing between levels, seeing across multiple servers on the Internet, and refractive water.

Tremendous Extensibility

Most of Unreal's subsystems are designed to be highly extensible, by defining an abstract base class (such as URenderDevice[?] for a general rendering device) to expose the general functionality to other modulars, while implementing one or more specializations of the subsystem (such as UGlideRenderDevice[?] for the 3dfx rendering code, or UOpenGLRenderDevice[?] for the OpenGL[?] rendering code). The following are some of the modules which are designed for extensibility:

- Actor Modules
- Game Types
- Rendering Modules
- Sound Modules
- Data Import Modules
- Data Export Modules
- Console Modules
- Network Protocol Modules
- Network Helpers
- Way More

User interface & localization

- Object-oriented, highly expandable in-game menu system for running in fullscreen-mode.
- Customizable, hierarchical subsystem-oriented "preferences" setting user interface in-a-window.
- Native support for localization of text to 8-bit languages, via CODEPAGE 850 and replaceable fonts.
- Built-in UnrealScript[?] and C++ support for externalization of all text, enabling non-programmer translation to all 8-bit languages.

Deliverables provided to licensees

- UnrealEd and the standalone .DXF importing tools.
- Full, commented C++ and UnrealScript[?] source code for everything.
- All corresponding executables and data.
- Limited documentation.
- All Unreal 1 game-specific content and game scripts. (For example purposes only. Not redistributable.)
- Updates for the duration of the license period, provided at milestones during development, which is on average every one to two months.

Licensing & business information

Contact Mark Rein: mrein@epicgames.com.