

search

Get the latest Education e-news [sign up](#)[Home](#)[Features](#)[Schools](#)[Digital Counselor](#)[Student Games](#)[Jobs](#)[Forums](#)[Contact Us](#)[Home](#) > [Features](#) > [What is a Game Engine?](#)

FEATURES

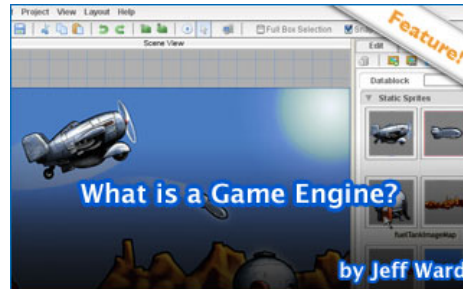
What is a Game Engine?

[04.29.08]

- Jeff Ward

If you follow the game industry, you'll hear the term "game engine" thrown around a lot. And I bet sometimes, late at night when everyone else has gone to sleep, you sit and wonder, "What is this nebulous thing I keep hearing about?"

You would expect that the answer would be as simple as being shown a car's engine: "Yeup, thar she is." After all, the game engine, much like a car's engine, is what makes the game go. Unfortunately, sometimes there's a fuzzy line between where a game's engine ends and where the content of a game begins, as if there were a fuzzy line between whether a car's air conditioner is part of its engine.



Generally though, the concept of a game engine is fairly simple: it exists to abstract the (sometime platform-dependent) details of doing common game-related tasks, like rendering, physics, and input, so that developers (artists, designers, scripters and, yes, even other programmers) can focus on the details that make their games unique.

Engines offer reusable components that can be manipulated to bring a game to life. Loading, displaying, and animating models, collision detection between objects, physics, input, graphical user interfaces, and even portions of a game's artificial intelligence can all be components that make up the engine. In contrast, the content of the game, specific models and textures, the *meaning* behind object collisions and input, and the way objects interact with the world, are the components that make the actual game. To use the car analogy again, think of how the body, CD player, in-dash navigation system, and leather seats make the actual car. That's the content.

APIs and SDKs

Two other terms you hear in the game industry that are closely related to game engines are "API" (application programming interface) and "SDK" (software development kit). APIs are the software interfaces that operating systems, libraries, and services provide so that you can take advantage of their particular features. An SDK is a collection of libraries, APIs, and tools that are made available for programming those same operating systems and services. Most game engines provide APIs in their SDKs. The Unreal Engine, for example, provides an interface for programmers to create their games, both through a scripting language called UnrealScript, and through libraries, which are provided to anyone who licenses the engine, and which come in the same package as their other tools, like the editor UnrealEd.

But where do game engines come from?

Next: [Continued](#)[Printer-Friendly Version](#) Pages: 1 2[permalink](#)

Comments