Games Engines – Current Offerings, Comparisons and Research

Games engines are the core software component for video games. the engine encompasses the game logic, the graphical rendering system, the AI (Artifical Intelligence), Physics, input system, sound and music, as well as the scripting system for movement of the characters in game as well as story progression. The graphical rendering engine involves such features as lighting, texturing, shadows, in most modern engines: shaders, rendering and scene management. The physics engine is divided into two types: real time and high precision. These are able to produce basic physics for people, bodies and vehicles. The physics engine also handles collision detection with a number of different methods: distance based, intersection based, line object-interaction and object-object interaction. The AI engine deals with giving emotion to the NPCs (Non Player Characters), path finding, decision making, scripting and the finite state machine. Due to the increasing complexity of such engines, many developers (both commercial and hobbyists) choose to use a currently existing engine and modify it for their idea, rather than develop their own engine, this decreases development time and effort. There are some websites set up specifically to enable this process to be easier. [1]. The amount of different games engines is almost overwhelming with most original game engines, still partially in use today, stemming from the early 1990s with Doom and then Quake. These have then been extended and reworked to produce engines such as the Unreal engine and Source engine which will all be examined further on. These engines will be examined followed by a look at MMOG network architecture and a comparison hereafter.

History and round up of the most popular game engines

Doom is often referred to as the original FPS (First Person Shooter) by fans who assume it was, although it was actually Wolfenstein 3D with that title [2] with Doom being the original FPS with multiplayer capability. [3] The Doom engine source code is now freely available under the GNU General Public License (as of 1997) [4], but was originally used for games such as Hexen and Doom 2. The release of the source code also enabled many Doom source ports to be created by fans of the original game. The next pivotal step in FPS games engine was again from id with Quake and the Quake engine and then its sequel Quake 2. Both of these engines were also released under the GNU General Public License in 1999 and 2001 respectively. [5] Quake 2's engine was used for games such as Kingpin, Daikatana, Sin and Soldier of Fortune [5]. Quake 3 produced similar results with its engine spawning Star Trek: Elite Forces and Return to Castle Wolfenstein. [5].

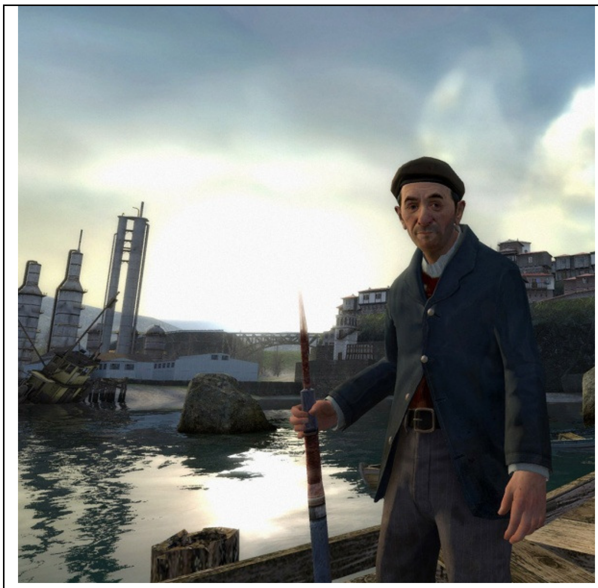
The Unreal Engine developed by Epic Games has produced similar results with each new Unreal game spawning a new engine with added features such as improved physics and graphics. The Unreal engine has even been proven to be so flexible as to be ported from PC to PS2 for games such as Deus Ex: The Conspiracy [6]. The UnrealEd level editor has attracted praise from a variety of sources, it is used by Universities on courses such as Staffordshire's Bsc Computer Games Design [7] as well as by a variety of 'modders' [8]. This is arguably because of its intuitive and

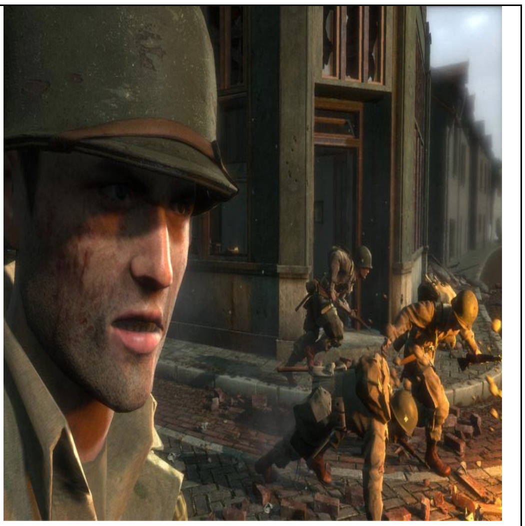
CAD like interface. As Vito Miliano discusses [9], the Unreal Level Editor can even be used for commercial projects such as providing the tour of a house for sale online rather than having to physically be there. Games engines can be used for a variety of things other than the primary idea of games. The Unreal Engine 2 was used primarily for Unreal Tournament 2004. It supports DirectX8 and OpenGL. [10] There is also an Unreal Engine 2X which is an optimised version of the engine for the X-Box. [10] The Unreal Engine 2 has also been used for Splinter Cell, America's Army and even the MMORPG Lineage 2, which demonstrates the flexibility of the engine. [11] The engine offers BSP Geometry, static meshes, animated skeletal meshes, vertex meshes and height-mapped terrain. The Unreal Engine 2X offers similar features as well as full scene light bloom and emissive materials. It also supports two different modes of depth of field. 'The first mode is always on, is based upon an absolute depth/blur amount, and enhances z-depth perception. The other mode is used to focus on specific objects in the scene, blur objects outside the focus depth planes, and give the player the feel of looking through a rifle scope.'. [12] A feature specifically relevant for console gamers is it offers dynamic gamma correction ensuring that the player never has the problem of their television screen becoming too dark in a crucial moment.

These concepts have since been extended to the Unreal Engine 3 which has been designed with DirectX9 and consoles such as the X-Box 360 in mind. This engine has been used so far for games such as Gears of War on the X-Box 360 and the forthcoming Unreal Tournament 2007. [13] The engine also encompasses the use of the AGEIA PhysX Processor which is reported to vastly improve the physics engines within games according to its manufacturer, AGEIA. [14] The engine supports graphical features such as HDR (High Dynamic Range) Rendering which is mainly available on PCs with relatively high specification graphics cards produced by either ATI or Nvidia. [15] The engine supports a rigid body physics engine that optimises collision detection. It also supports a skeletal animation system enhancing the realism of the models' movements, as well as improved AI and network architecture, amongst other things. [16] The engine has since been licensed to Microsoft, Sony (for the PS3), EA and a number of other next generation developers. [17] The development of Unreal Engine 4 is also underway but with no signs of its arrival for a number of years yet. [17]

The other core games engine used for a number of different games, again predominantly FPS games, is the Source engine devised by Valve Software. This engine carries on from the work Valve implemented for Half Life. The Source engine is used for Half Life 2 and various mods, such as Counterstrike: Source and Day of Defeat: Source, as well as spin off games such as Half Life 2: Episode 1 and Half Life: Source[18]. The engine offers a wide number of features, some similar to the Unreal Engine's offerings.[19]

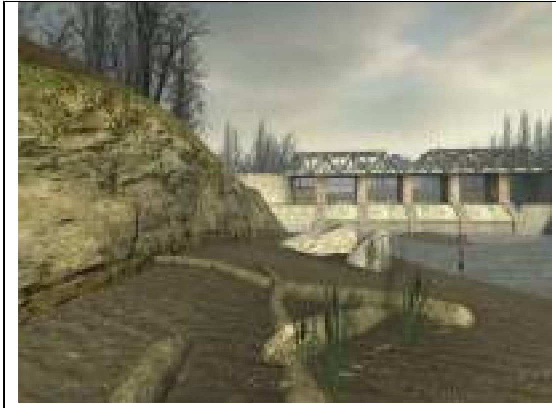
It offers shaders 2.0 (much like the Unreal Engine), dynamic lighting and HDR lighting (again, like the Unreal Engine); it also provides a wide range of scalability, allowing the engine to be used on PCs with DirectX6.0 technology as well as up to DirectX9.0 technology. It also uses an advanced physics engine, perhaps best highlighted with the usage of the grav-gun in Half Life 2: weapon that allows the player to pick up any item and throw it anywhere. The game also allows interaction with parts of the scenery such as barrels, which the Unreal Engine does not currently do so. It also has enhanced vehicle dynamics with realistic suspensions with springs on each wheel. It also offers advanced character detailing such as realistic looking eyes and simulated musculature. Below is an example of the difference in character detailing between the two engines:

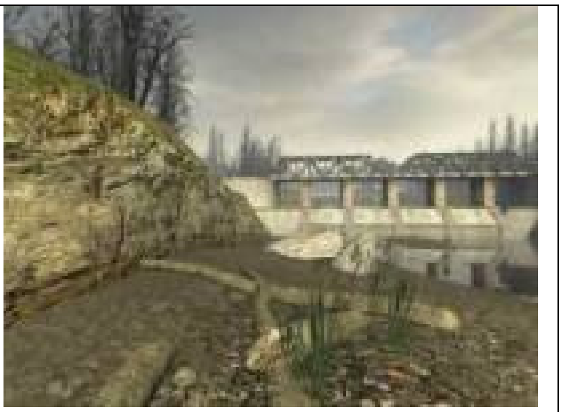

Source Engine in action[19]


Unreal Engine 3 in action [20]

Unlike the Unreal Engine, the Source Engine has recently added multi-thread support to its engine, giving owners of dual-core and quad-core CPUs an advantage with Source Engine games regarding processing capability.[21] It is hoped that this could be a big change in the development of games with its potential to allow developers to do more with the engine, without any concerns over ‘giving the CPU too much to do at once’. It will also hopefully decrease rendering time enabling a smoother game playing experience, as well as improving AI and path-finding amongst NPCs, making NPCs more unpredictable for the player. These changes are hoped to be implemented within Half Life 2: Episode Two due for release at the start of 2007.

The Source Engine with its increased flexibility also brings about how non Engine based software such as Microsoft’s DirectX can affect how games are produced and displayed. The Valve Developer’s Wiki [22] demonstrates the difference in graphical quality depending on which version of DirectX is used. Each version has gradually added more features, with the difference between DirectX6.0 and DirectX9 being depicted below:


DirectX6.0


DirectX9.0

DirectX9.0 improves on 6.0 by offering:

- Blob shadows
- Displacement map texture blending
- Refractions with the use of a bump-map
- Softer edge dynamic shadows
- Directional lighting on world brushes using normal maps
- Improved-quality specular effects
- High-quality reflective water (used frequently)
- Normal-mapped lighting on models
- High dynamic range rendering

There are other less dominant, but equally important engines in use for a variety of games. One of the most flexible is that of the RenderWare engine. This engine has been used by Criterion for a number of games such as Grand Theft Auto: San Andreas, Call of Duty: Finest Hour and Burnout 3: Takedown. [23] The engine is available for all the current generation of consoles as well as the PC. It is evidently a flexible engine with its ability to be used for a high speed racing game such as Burnout 3: Takedown and also the point and click adventure game Broken Sword: The Sleeping Dragon. The RenderWare Platform itself is technically middleware: Middleware being 'the software layer that lies between the operating system and the applications on each site of the system'. [24] There are also a wide range of more independently geared games engines such as Garage Games's Torque Engine which is available to anyone for $150 [25], and also the MultiVerse Engine which caters specifically for the MMORPG (Massively Multiplayer Online Role Playing Game) market. [26] This engine uses a combination of C# and Python[27] to implement its game features, a sharp contrast from the Source and Unreal Engine's reliance on C++. These engines are mainly aimed for the smaller developer rather than larger companies, although the Torque Engine has been used for games such as Marble Blast Ultra on the X-Box 360 which has been released on X-Box Live Arcade, as well as Tribes 2. [28] In many cases it is arguable that it would be better for budding 'bedroom' games developers to concentrate on using an established engine such as the Source Engine so as to reach a larger audience.

To a lesser extent there are also isometric engines, used primarily for game genres such as RTS (Real Time Strategy) games and RPGs (Role Playing Games). These give a partly 2D/partly 3D perspective of the action from an overheard view. Examples of these types of games would be Diablo 1 and 2 and Baldur's Gate 1 and 2. They tend to look less realistic but it enables the PC or console to display a much larger amount of information, such as an army of 500, than if it had to display each and every character in an accurate and realistic fashion.

Examination of MMOG engines and network architecture

Different concepts need to be taken into consideration when evaluating a suitable engine for a MMOG, most notably dealing with the network architecture required. This is mainly done through distributed architecture, either through the cluster-based or zone-based approaches. The cluster-based approach is where clients are connected to a specific cluster according to network topology, while the zone-based approach is where clients are connected to a specific zone-server corresponding to the geographic area the player wishes to join within the game. Both approaches have their own advantages. Cluster-based architecture reduces latency through bandwidth dynamic adaptation, while zone-based architecture means only part of the game world needs to be replicated at any one time for the player, meaning it can be quicker for the player.

There are also three main strategies for reducing the bandwidth requirements of MMOGs. To reduce the amount of hosts information is sent to, there are AOIM (Areas of Interest Management) filtering techniques which can be implemented. There is also the potential possibility of using P2P (Peer to Peer) 'overlays' to support a MMOG rather than using a centralised server. This is still theoretical however but has been discussed by Abdennour El Rhalibi, Madjid Merabti and Yuanyuan Shen [29] and Caltagirone, Keys, Schlief and Willshire. [30] To reduce the number of messages sent, aggregation strategies can be implemented, causing reduction on the size of the headers of the messages, compression strategies can also be implemented to reduce the size of the messages. Delay compensating techniques can also be used. These can mask common latency issues, as well as impact gameplay and cheating possibilities, through a number of techniques. Client Side Extrapolation works by predicting what is most likely to happen and rendering that event. This technique can cause some inconsistency in the game state due to its predictions. Client Side Interpolation is where a past state is always rendered. 'Objects that are not controlled by the client are represented where they were at an amount of time equal to the estimated latency.' [31] The final most commonly used technique is server side latency compensating. This technique consists of 'evaluating each player action in the exact context in which they occurred' [31] and using the client game state of that moment as a reference combined with an evaluation of the client's latency at the same time.

The Source Engine, as summarised earlier, deals with network architecture issues in a similar way as most MMOG engines. The client machine communicates with the centralised game server to evaluate the moves made within the game. It does this through data packet communication, however it is crucial to be aware of how limited bandwidth is. The server is unable to send a new update packet to all clients for every single movement or world change. Instead it takes snapshots of the current world state at a constant rate and sends those snapshots to the clients. It takes a certain amount of time so the clients are always slightly behind the server with updates, packets of data can also get lost in the transmission, a process known as 'packet loss'. This can disrupt how accurate shots are fired within a fast paced FPS game such as Counter-Strike. The Source Engine aims to combat this by using similar techniques as MMOGs such as data compression, interpolation, extrapolation and lag compensation. [32] The Counter-Strike Source Server simulates the game in discrete time steps known as 'ticks'. The server software then determines whether clients need a world update according to what changes in that 'tick'. A higher tick rate means more updates but will also increase CPU load on the server. Lag compensating techniques are especially crucial for such a fast moving FPS game, but they are still necessary in any online game so as to be able to offer the game at its best even when dealing with potential internet related issues such as high pings or packet loss.

Conclusions

The issues regarding modern day game engines are becoming increasingly complex. With an increasing number of FPS games also needing a strong multiplayer facility within their games, network architecture and lag compensating techniques are also crucial to a game's engine and its potential success once released. Currently it is arguable that the Source engine is slightly ahead regarding efficient net code but it is behind when it comes to graphical power, with the Unreal Engine 3 being the favoured engine for consoles such as the X-Box 360 which uses the engine for games such as Gears of War. Valve, developers of the Source Engine, is currently working on an optimised Source Engine for the 360. [33] Stang argues that game engines are becoming more accessible to non-commercial enterprises with the improved capabilities of engines such as the Torque Game engine and the reduction in cost to acquire these engines. [34] This may not be entirely beneficial however as it would still take some considerable talent to be noticed through these engines, with possibly constructing an Unreal or Source Engine based mod being more noticeable due to their extensive modding communities. The future of games engines will most likely involve the dominance of a few commercial engines such as the Unreal Engine and Source Engine with them both continuing to improve upon current features such as the physics and graphical rendering engines. The net code will also most likely be honed to make it more efficient alongside the increase in broadband speeds offered to the public.

References

[1] DevMaster, 2006. [online] Available from: http://www.devmaster.net/ [Accessed 29th November 2006]

[2] The Gamasutra Quantum Leap Awards: First-Person Shooters, 2006. Gamasutra [online] Available from: http://www.gamasutra.com/features/20060901/quantum_01.shtml [Accessed 29th November 2006]

[3] Tracing Trends: Multiplayer FPS, 2005. Bit Tech [online] Available from: http://www.bit-tech.net/gaming/2005/12/08/multiplayer_fps/1.html [Accessed 29th November 2006]

[4] 3D Downloads, 2001. [online] Available from: http://www.3ddownloads.com/showfile.php3?file_id=7430 [Accessed 29th November 2006]

[5] ID Software's Technlogy Licensing Program, 2001. ID Software [online] Available from: http://www.idsoftware.com/business/technology/techlicense.php#endnote2 [Accessed 29th November 2006]

[6] Deus Ex: The Conspiracy for Playstation 2, 2001. Deus Ex Machina [online] Available from: http://www.deusex-machina.com/dxps2/interviews/interview1.asp [Accessed 29th November 2006]

[7] Duckworth, T. 2006. Personal Communication

[8] Planet Unreal, 2006. [online] Available from: http://www.planetunreal.com/ [Accessed 29th November 2006]

[9] Miliano, V. 1999.
Unrealty: Application of a 3D Game Engine to enhance the Design, Visualization and Presentation of Commercial Real Estate. [online] Available from: http://www.unrealty.net/vsmm99/ [Accessed 29th November 2006]

[10] Unreal Technology. [online] Available from: http://www.unrealtechnology.com/html/homefold/home.shtml

[11] Unreal Engine 2, 2003. Unreal Technology. [online] Available from: http://www.unrealtechnology.com/html/technology/ue2.shtml [Accessed 29th November 2006]

[12] Unreal Engine 2X, 2004. Unreal Technology. [online] Available from: http://www.unrealtechnology.com/html/technology/ue2x.shtml [Accessed 29th November 2006]

[13] Gears of War, 2006. [online] Available from: http://www.gearsofwar.com/about/ [Accessed 29th November 2006]

[14] PhysX by Ageia, 2006. [online] Available from: http://ageia.vnewscenter.com/press.jsp?id=1141898679428 [Accessed 29th November 2006]

[15] Unreal Engine 3, 2004. Eurogamer. [online] Available from: http://www.eurogamer.net/article.php?article_id=55942 [Accessed 29th November 2006]

[16] Unreal Engine 3, 2005. Unreal Technology. [online] Available from: http://www.unrealtechnology.com/html/technology/ue30.shtml [Accessed 29th November 2006]

[17] Computer and Videogames.com, 2005. [online] Available from: http://www.computerandvideogames.com/article.php?id=123639 [Accessed 29th November 2006]

[18] Original Half Life to be released in Source Engine, 2004. Gamespy. [online] Available from: http://uk.pc.gamespy.com/pc/half-life-2/515917p1.html [Accessed 29th November 2006]

[19] Source Engine Features, 2006. The Valve Developer Community. [online] Available from: http://developer.valvesoftware.com/wiki/Source_Engine_Features [Accessed 29th November 2006]

[20] Blick Online, 2006. [online] Available from: http://www.blick.ch/games/news/artikel35173 [Accessed 29th November 2006]

[21] Multi-Core in the Source Engine, 2006. Bit-Tech. [online] Available from: http://www.bit-tech.net/gaming/2006/11/02/Multi_core_in_the_Source_Engin/1.html [Accessed 29th November 2006]

[22] DirectX Versions, 2005. The Valve Developer Community. [online] Available from: http://developer.valvesoftware.com/wiki/DirectX_Versions [Accessed 29th November 2006]

[23] Renderware, 2005. [online] Available from: http://www.renderware.com/ [Accessed 29th November 2006]

[24] What is Middleware?, 2006. ObjectWeb. [online] Available from: http://middleware.objectweb.org/ [Accessed 29th November 2006]

[25] Torque Game Engine, 2006. Garage Games. [online] Available from: http://www.garagegames.com/products/torque/tge/ [Accessed 29th November 2006]

[26] Multiverse, 2006. [online] Available from: http://multiverse.net/ [Accessed 29th November 2006]

[27] Multiverse Technology: An overview, 2006. Multiverse. [online] Available from: http://multiverse.net/platform/whitepapers/mv_overview.pdf [Accessed 29th November 2006]

[28] Commercial Games, 2006. Garage Games. [online] Available from: http://www.garagegames.com/solutions/games/ [Accessed 29th November 2006]

[29] El Rhalibi, A., Merabti, M., and Shen, Y. 2006. AoIM in peer-to-peer multiplayer online games. In *Proceedings of the 2006 ACM SIGCHI international Conference on Advances in Computer Entertainment Technology* (Hollywood, California, June 14 - 16, 2006). ACE '06. ACM Press, New York, NY, 71.

[30] Caltagirone, S., Keys, M., Schlief, B., and Willshire, M. J. 2002. Architecture for a massively multiplayer online role playing game engine. *J. Comput. Small Coll.* 18, 2 (Dec. 2002), 105-116.

[31] Bosser, A. 2004. Massively multi-player games: matching game design with technical design. In Proceedings of the 2004 ACM SIGCHI international Conference on Advances in Computer Entertainment Technology (Singapore, June 03 - 05, 2005). ACE '04, vol. 74. ACM Press, New York, NY, 263-268.

[32] Source Multiplayer Networking, 2004. Valve Corporation. [online] Available from: http://www.valve-erc.com/srcsdk/general/multiplayer_networking.html [Accessed 29th November 2006]

[33] Valve's Source Heads to X-Box 360, 2006. Next Generation. [online] Available from: http://www.next-gen.biz/index.php?option=com_content&task=view&id=4018&Itemid=2 [Accessed 29th November 2006]

[34] Stang, Bendik. 2003. Game Engines: Features and Possibilities. In IMM DTU 2003 (Denmark, 2003).