

The Evolution of Game Engines and their Licences



By Steven Walsh



NUI MAYNOOTH
Ollscoil na hÉireann Má Nuad

Table of Contents

Abstract	3
Introduction	4
SCUMM at the Start.....	5
The DOOM of Adventure Games	6
The Quake felt around the world	8
Unreal competition.....	10
Quake at its Source	12
Tech Demo to Game Engine	13
Can it play Crysis?	15
A Newcomer and an old Master	16
Conclusion	18
The Middleware for Middleware	18

Abstract

In this report I will discuss the evolution of game engines, their licenses', legacy, and impact on modern day game development, as well as gaming as a whole. As this is a report on the *evolution* of game engines I will approach this topic chronologically, detailing the rise in prominence of some of the most influential game engines. Although every game runs on an 'engine' this report will only look at the game engines made with the intent of licensing to other developers and even within these engines I will only be looking at the most widely used, interesting(which will be decided by me), and novel ones. With the growth of game engines as a viable business there was also companies who started developing middleware plugins to be used in conjunction with game engines, although technically game engines themselves are middleware but we won't quibble, and I will discuss the most popular of these middleware applications as well.

I was not able to obtain any solid data on the licencing of game engines. I emailed several companies to enquire about licencing but no information could be divulged. I could not find any direct figures for licencing costs but some references have been useful to shed some light on this seemingly lucrative business.

I will also be mentioning important games, developers, and individuals throughout this report to give more information on the engine itself or the background behind engine.

Introduction

What is a game engine? The easiest way to describe a game engine is to compare it to a car's engine. Both forms of engine make what they power go. In both cases there are many engines to choose from, all with different capabilities, and some suited to certain jobs better than others. But they all make things go. No one would of course buy a car engine to drive around, companies, who either purchases engines or make their own, build cars around these, just like games are built around game engines. Two cars could have the exact same engine, but look and drive completely differently, as engines are the very bones of what is needed and, apart from safety equipment, all the rest is mostly aesthetic. This is exactly the same of game engines. Car engines are mechanical in nature and come as big lumps of metal, pistons and bolts, whereas game engines come in application programming interfaces (API's), software development kits (SDK's), and sometimes editor tools. Game engines provide a framework and libraries ready to use so a developer can spend more time on creating games rather than the tedious intricacies of making them work. Game engines often include libraries for multi-platform development, networking models, memory management, rendering 2D and 3D data, AI, and collision detection to name a few.

Every game needs an 'engine' to run on but in the early years, when the gaming industry was in its infancy and technology was limited, it wouldn't take much manpower or time to create an engine for a game, so developers made all the engines for their games in house, rarely reusing the same engine, and it was unheard of for people to sell game engines. As technology advanced, it took more time and greater manpower to create sophisticated game engines to get the most out of the hardware. Companies who specialised in certain genres of games would reuse a lot of code to cut down on development time and costs. During the nineties the process of selling or 'licensing' game engines became popular. We will get into the details later but the main influence for this was the game DOOM made by id software. DOOM was released as shareware in 1993 and was commercially available in stores in 1995. In late 1995 DOOM was estimated to be installed on more computers than Microsoft's latest operating system, Windows 95. This caused Microsoft to work on a Windows port for DOOM and used it to advertise Windows 95 itself¹. Other developers wanted to emulate the success of DOOM and its amazing graphics for the time, and some decided to license the game engine running it. Licenses can either be a one off payment or

¹ <http://www.youtube.com/watch?v=uJdCOpUJPIA>

work on a royalty based system with a percentage of the profits going to the licensor. Id software has been at the forefront of game engine development since then and continues to this day.

SCUMM at the Start

During the mid to late eighties the biggest selling and most popular games were adventure games. These games would not give direct control of a character's actions by pressing buttons but by giving a list of words at the bottom of the screen. These words would be nouns, adjectives, and phrases such as 'Walk to' and 'Pick up'. Players would string together these words in certain situations to solve puzzles such as walking to a part of the room, picking something up, travelling somewhere else, and using that item to open something. These games required quite complex text parsers and as the options grew the complexity grew as well. The two biggest developers of adventure games at the time were Sierra On-line and Lucas Arts. Both developed their own scripting languages to make it easier and quicker to produce adventure games. The scripting languages would have commands to draw scenes and interpret input from the user, which would be translated to libraries which the machine would understand and execute, allowing for more to be produced with less code and less time, much like a game engine. Sierra On-line had two scripting languages, Adventure Game Interpreter (AGI), and Sierra's Creative Interpreter (SCI), the latter being a newer version of the former. Lucas Arts created a scripting language for their 1987 adventure game Maniac Mansion, with the intent that it could be reused for future titles. It was named the Script Creation Utility for Maniac Mansion, or SCUMM for short, and was made by Aric Wilmunder and Ron Gilbert. Although a scripting language SCUMM is described as a crossover between a game engine and scripting language. Its design allowed for add-ons or plug-ins to be implemented within it, very much like a game engine. These add-ons included the animation engine, called FLEM, this placed names and objects inside a room, and CYST, this is used for character animation. SCUMM was an in house engine and as such was never licenced or sold to other developers. It was used within Lucas Arts for thirteen games and until it was replaced by the scripting language Lua in the game 'Monkey Island 2: Le'Chucks Revenge'.

SCUMM games are not compatible with newer operating systems and Lucas Arts has provided little to no support to help overcome this feature. In the 2000's a internet community started up something called the SCUMMVM, or SCUMM Virtual Machine, to enable people to be able to play the old SCUMM games on newer machines. This virtual machine acts as an emulator to the game so the game is 'mounted' or played through the virtual machine. This allows it to work on newer systems.

The DOOM of Adventure Games

"I blame DOOM. That Game showed up and interjected testosterone in gaming that wasn't there before and adventure games had a hard time competing with that energy"², this is a quote from Ron Gilbert, creator of SCUMM and lead designer for Lucas Arts adventure games, on the effect on DOOM. DOOM was a milestone when it was released. It gave the illusion of full 3d graphics but was made up of 2d sprites arranged to mimic 3d structures. This kind of design was called 2.5d. DOOM was created by id software in 1993. It was not the first game to use the 2.5d perspective, and not even the first from id software, but the engine created to run DOOM was ground breaking. Id had created similar engines for previous games, all adding something new. The engine for the game 'Hovortank 3d'(1991) had solid colour drawn walls, 'Catacomb 3d'(1991) introduced texture mapping, and the engine for 'Wolfenstein 3d'(1993) introduced 256-colour VGA instead of the old 16-colour EGA. The engine for Wolfenstein 3d was licenced to make several other games. The most appealing feature about the Wolfenstein engine was that it ran very well on weak hardware, it didn't have some features that other engines did, but more people being able to play a developer's game was appealing just as much as better technology. The 3d effect of these early id software engines was created by ray casting. This technique works by emitting one ray for each column of pixels, the engine then checks if the ray intersects a wall, walls would be given defined positions before had, and then it draws the texture or colour in the point of the ray intersecting the wall. The programmer John Carmack is the lead programmer for id and has produced every game engine since the company's inception.

Id software was formed in 1991 when four employees, John Carmack, John Romero, Tom Hall, and Adrian Carmack, at computer company Softdisk decided to create their own games company. John Carmack was working on a scrolling engine for the PC to

² <http://www.rockpapershotgun.com/2008/01/12/rps-exclusive-ron-gilbert-interview/>, Ron Gilbert, 12/01/2008

emulate games like Super Mario from the Nintendo entertainment system. It was hard to get effective scrolling for platform games like Super Mario on PC's at the time due to EGA. John Carmack created an engine that achieved this and when showing it to colleague John Romero, Romero stated "What I had in front of me was the future"³, an excerpt from his personal blog. John Carmack had recreated the first level in 'Super Mario Bros 3' with a different character named Dangerous Dave. The demo was called 'Dangerous Dave in: Copyright Infringement' and it's the game that created id software. John Carmack is world renowned for his achievements in computer graphics. He has pioneered a lot of technologies including ray casting, binary space partitioning⁴, surface caching, and Carmack's reverse, which is also known as z-fail stencil shadowing⁵ but is well known by Carmack's reverse after the release of DOOM 3.

Carmack created the game engine for DOOM which included all the features in the old engines, most of which was improved upon, and also introduced binary space partitioning. DOOM is credited as the first game to use this feature and it allowed it to render each room in the level quickly but at different times which meant more processing power could be directed to gameplay than to rooms which the player was not even in yet. It had notable improvements to previous id games including, full texture mapping, height differences, non-perpendicular walls, stereo-sound, dynamic levels i.e. moving platforms, and varying light levels allowing for greater atmosphere by putting the player in darkness. It was all these technologies combined that made DOOM a major success. The fast gameplay and great graphics was only possible by the technology behind the game. The game engine was powerful but also allowed the game to be run on weaker machines than a game of this standard would usually support. There was great demand for the DOOM engine and it was licenced to several other developers. Id released DOOM 2 in 1994 which was also on the DOOM engine, but with several improvements made, id didn't make another wholly new engine until 1996 with Quake. Id software are great supporters of the open source community and release their source code for previous engines under the GNU General Public License after a newer one has been released the source code for DOOM can be found on id software's github page⁶.

³ http://rome.ro/games_ddici.php, John Romero

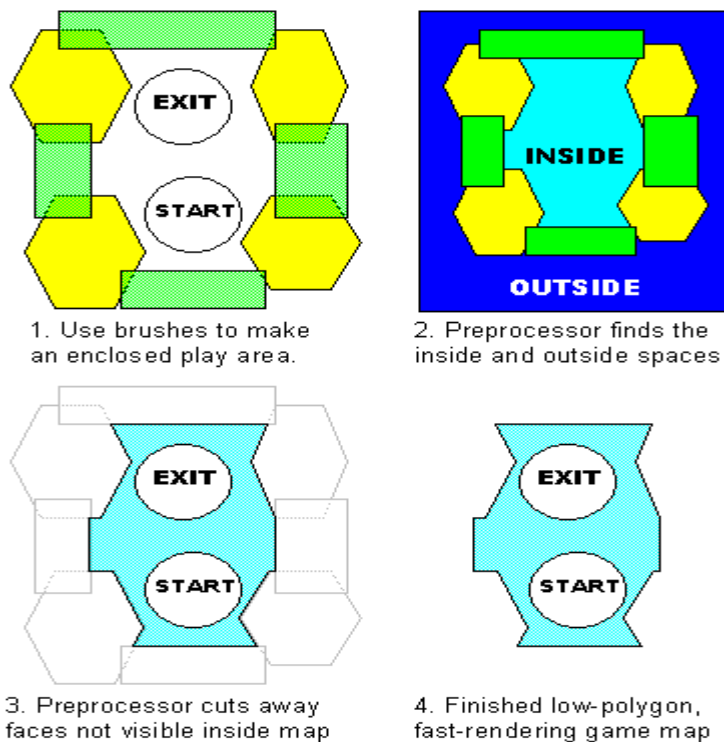
⁴ <http://maven.smith.edu/~mcharley/bsp/>

⁵ <http://maverick.inria.fr/Publications/2005/HHLH05/hhlh-zp-plus.pdf>

⁶ <https://github.com/id-Software/DOOM>

The Quake felt around the world

In 1996 id software released Quake with a new engine powering it. Quake was a fully 3d game instead of 2.5d as previous id software games. It also included a full TCP/IP networking system which was developed by Carmack and newly hired employee Michael Abrash, who was famous for his “The Zen of Assembly Language”⁷ books. In a recent blog post by Abrash⁸ he talks about the development of Quake and its engine and the new technologies introduced by himself and Carmack. Quake popularised polygonal models and 3d environments. At the time of release computers were still not very powerful and standard computers would struggle with rendering such complex games. A special map design system was created using vertex-pruning to pre-process and pre-render the environments to reduce processing required to play the game. This was done by using polygonal brushes to make a play area, the pre-processor would determine the outside and inside play space, it would prune out the vertices not visible from the inside play space to give a low level render of the basic environment with textures, models, and lighting added in later when loading the game.



⁷ <http://www.federaljack.com/ebooks/Computers%20-%20Informatin%20Technology/Assembly%20,%20reverse%20engeniering%20and%20Computer%20Architecture%20books/Assembly/Before/various/The%20Zen%20Of%20Assembly%20Language%201990%20-%20Michael%20Abrash.pdf>

⁸ <http://blogs.valvesoftware.com/abrash/>, Michael Abrash, 13/04/2012

engine and not much has been changed in the way 3d games are implemented. The Quake 2 engine directly inherited from the original Quake engine whereas the Quake 3 engine was written from scratch but still inherits the 3d properties and techniques developed from the Quake engine. One licensee to note is Valve Software who licenced the original Quake engine for their game Half-Life. Valve was founded by Mike Harrington and Gabe Newell, both having previously worked at Microsoft. They started their own game company and decided to use the quake engine as the base for their new game. In Abrash's blog, the same one mentioned earlier, he states he helped them obtain the licence as he knew Harrington and Newell from previously working at Microsoft. He also states that "Id made a lot of money from that license", Half-Life had sold close to ten million copies as of 2006.

Quake 2 was released in 1997 and featured an updated engine. The basic principles behind the Quake 2 engine were the same as the previous Quake with the main. The main improvements made on the engine were support for hardware-accelerated graphics, specifically OpenGL, as well as the propriety software renderer. The engine code was also sub divided into dynamic-link-libraries which enabled both the software render and the OpenGL render to be included. If the computer playing the game had support for OpenGL it could be easily implemented. It also made it possible to distribute the code for the modding community without releasing proprietary parts of the engine itself. The change to OpenGL was possible with the release of dedicated video cards for computers. The original Quake was also released with OpenGL support in 1997 and both Quakes OpenGL features are based on the 3Dfx based video cards of the time. Quake.ie, a fan site set up for all things Quake, states that "The release of GLQuake(as in 'OpenGL' Quake) in January of 1997 made 3Dfx the winner of the first video card hardware war, and Carmack's choices would affect how drivers for such hardware were written for years"¹⁰.

Unreal competition

In 1998 Epic MegaGames, now known as just Epic Games since 1999, released Unreal. Epic had been making games since the early 90s but had never licenced a previous games engine before Unreal, aptly named, the Unreal engine. The Unreal engine was in direct competition to the Quake 2 engine, which had a monopoly prior to Epic releasing their engine. The Unreal engine is considered technically superior to the Quake 2 engine. The Unreal engine made

¹⁰ <http://www.quake.ie/about>

great strides in engine technology which made it stand above the Quake 2 engine. The most notable feature was detail texturing which worked by allowing multiple textures applied to one object. The closer the player got to the texture a more detailed texture would fade in, giving a better picture where normally up close most textures would turn blurry. Other features that made Unreal popular was its self-contained scripting language, UnrealScript. It is similar in design to Java and is object-oriented. UnrealScript made it quick and easy for people to create mods(modifications) for games using the Unreal engine, as anything written in the script would be compatible within the engine and certain features could be called upon using a lot less code than an actual programming language. Unreal was also the first engine to use a client-server model for multiplayer gaming.

Quake 3: Arena was released in 1999 along with a newer more powerful engine. The Quake 3 engine, unlike the Quake 2 engine, was not based on a previous one but rather redone from scratch. The main difference from previous iterations of id software engines was that it no longer contained its own software renderer and a computer must have an OpenGL compliant graphics card. Both the Unreal and Quake 3 engines were different than previous ones in the fact that the engine and games they were based on were developed separately. This helped make the engines more portable to licensees. Both engines have been extensively licensed and used up until the newer releases of their company's engine, with the exception of the Quake 3 engine. Since being released under the GNU General Public License it is still being used today for modifications and developing new games, although it has seen less use since the DOOM 3 engine, its successor was released under GNU in 2011. On John Carmack's blog he stated that "I intended to release the Q3 source under the GPL by the end of 2004, but we had another large technology licensing deal go through, and it would be poor form to make the source public a few months after a company paid hundreds of thousands of dollars for full rights to it."¹¹, which gives a little insight into the money available from licensing such an engine.

Unreal engine 2 was released in 2002 adding increased performance as well as support for the Nintendo Gamecube and the Xbox. It also integrated with the Karma physics SDK to provide ragdoll physics for Unreal based games. Unreal 2 was completely re-written from the original Unreal engine. It also began Epic's structure of versioning game engines with the release of Unreal 2.5 in 2004. Instead of waiting for a completely new engine Epic would update the

¹¹ <http://www.armadilloaerospace.com/n.x/johnc/Recent%20Updates>, John Carmack, 31/12/2004

original engine with new technology or improved functionality. The changes aren't as significant as a new engine but allowed Epic to keep their engine ahead of their competitors technologically with the updates. In 2004 Epic released Unreal 2.5 which introduced improved rendering as well as vehicle physics.

Quake at its Source

“There are still bits of early Quake code in Half-Life 2”¹². This is an extract from John Carmack's blog referring to Valve Software's game engine Source. In 2004 Valve released Counter-Strike: Source, debuting the Source engine. It was based off their own engine GoldSrc for Half-Life, created from the Quake engine. It featured cutting edge technology including High Dynamic Range Rendering, bandwidth-efficient physics engine, and dynamic lighting but its main feature was its animation system. Its facial animation system was ahead of its time giving characters almost life like movement and facial expressions. The facial expressions achieved with Source are still above standard for game engines today and are improved with each iteration of the engine. Source does not follow the traditional version numbering system as other engines but was designed to receive constant iterations for improvement. The benefit of this means that old games can be updated and make use of technology introduced in later iterations making sure that all of Valve's games are not out dated as newer technology presents itself. It also allowed a lot of source code to be viewed by anyone so that mods can be freely created with ease. Notable upgrades to the Source engine usually come with the release of a new game from Valve such as multiprocessor support with the release of the Orange Box. It also helps greatly with portability as in 2010 the Source engine was given Mac OS support which was then available for all Valve games using the Source engine.

2004 also saw a new release from id software, DOOM 3, and another engine to go with it. It was an extremely powerful engine with many new features. The stand out feature in the DOOM 3 engine was per-pixel lighting which dynamically rendered shadows and lighting. Previous engines used pre-calculated per-vertex lighting which meant that lighting and shadow for maps would be static and wouldn't change. This fully dynamic lighting and shadowing technique added a great atmosphere to the game. One criticism of the DOOM 3 engine was that it did not work well in large bright outdoor areas, and such areas

¹² <http://www.armadilloaerospace.com/n.x/johnc/Recent%20Updates>, John Carmack, 31/12/2004

would suffer a massive performance hit due to this technique, as DOOM 3 was based in door using dark environments. To solve this issue John Carmack developed a new texturing feature called MegaTexture, first used in the game Enemy Territory: Quake Wars by developer Splash Damage, who also licensed out the Quake IP. This was achieved by using a single massive texture, 32,768 x 32,768 pixels, covering the entire map and detailed terrain. This works by loading the entire texture into memory beforehand and streaming the required visible parts of the texture as the player moved around. This method increases loading time, and the textures are larger in size than normal textures, but it also uses up very little RAM for the size of the texture. This technique not only solved the engines problem but also made it one of the best engines for rendering such large outdoor spaces with the greatest detail. The DOOM 3 engine, despite its lighting innovations, was not as extensively licensed as previous game engines by id software or even other engines by other developers. Since MegaTexture was introduced the DOOM 3 engine has seen more licences than before despite being an older engine so this can be seen as a main turn off point for potential licensees.

Tech Demo to Game Engine

Crytek was a software developer founded in 1999 by the Yerli brothers, Cevat, Avni, and Faruk. They were hired by Nvidia to create a tech demo for their new graphics card. The demo was revealed at Nvidia booths in 2000 and was met with critical acclaim. After this Crytek released several demos of the game spawned from the tech demo, X-Isle, until it slowly evolved into their debut game Far Cry. They also renamed their engine to the CryEngine and started receiving requests from developers to licence their engine in 2002, before they even released a game. "To be honest, at the beginning that wasn't a priority for us. We just wanted to have a proprietary engine for our games projects,"¹³, the Yerli brothers never intended for their engine to be licenced but after receiving interest they decided it would be the best idea. In 2004 Far Cry was released to critical acclaim. The gameplay itself wasn't ground breaking, but not terrible, but the real achievement was the vast rendering capabilities of the engine. The game was set on an island, all of which could be explored. If you could see it, you could travel there. It featured long draw distances and no loading pauses between transitions to separate parts of the massive island, it also had seamless transitions between indoor and outdoor areas. It also had an advanced rendering system for vegetation which had some of the best vegetation in a video game for its time of release. All of these

¹³ <http://n4g.com/news/402863/develop-qandamp-a-the-yerli-brothers>

factors made the game engine stand out and was the engine of choice if massive areas needed to be rendered. In 2006 Crytek sold intellectual property rights of Far Cry and a perpetual licence to use the Far Cry edition of CryEngine to publisher Ubisoft, who have since released Far Cry 2 with an improved engine, named the Dunia engine, with Far Cry 3 due to be released within a year. Crytek also revealed the next game they are working on, called Crysis, and announced a new engine in development, CryEngine 2.

2006 also saw the release of a new engine from Epic, the Unreal 3 engine. It was released to use on the current generation of consoles, Xbox 360 and Playstation 3, as well as computers. Epic greatly pushed their new engine so it could be extensively licenced. Giantbomb.com, a massive online gaming community, have compiled a list of games using the Unreal 3 engine¹⁴, with maybe the possibility of some being missed, which come to one hundred and twenty four games using the engine. The engine saw more incremental updates than previous ones as new technology came along and also gained iOS and Android support in 2010, and Mac OS support in 2011.

In 2009 Epic released a free version of their engine, to everyone. It was called the Unreal Development Kit, or UDK, and was completely free to download and use. It did not have as many features as the fully licenced Unreal 3 engine, but quality games could still be made from it and was perfect to indie developers on a budget. It also publicly released its licencing requirements, which has not been done with other high end engines before¹⁵. A licence can be bought for 99 dollars. This licence allows you to release your games of software commercially. There is also a royalty fee paid to Epic, which is 0% on the first 50,000 dollars your software makes and 25% on any subsequent income. The UDK has become very popular with indie developers and modding communities due to its free use for such a powerful engine. The UDK also receives incremental updates as well as the Unreal engine, including compatibility for iOS, Android, and Mac OS. Around this time game engines were becoming very powerful. They became one of the largest and most complex software systems to produce and the detail that was achievable by them was put to use beyond games. Many engine developers started licencing their engines for simulation software or animation. One such licence from the Unreal 3 engine is the hit television show LazyTown¹⁶. In recent years governments have been increasingly interested in game engines

¹⁴ <http://www.giantbomb.com/unreal-engine-3/92-86/games/>

¹⁵ <http://udk.com/licensing>

¹⁶ http://www.unrealengine.com/showcase/film_television/lazytown/

to produce simulation software for armies, police, and even doctors. Since game engines were being able to produce games that simulated real physics very well, as well as other factors such as lighting and such, governments would licence the software to make simulations. One of the biggest areas for simulation software is military simulation as it is much safer than real world. In spring 2012 Epic licenced their engine to the FBI and other offices for development of simulation software¹⁷. The Unreal engine has been used in what is called a military simulation before, America's Army, but AA could be played by the public, whereas these simulations will not be released.

Can it play Crysis?

In 2007 Crytek released their new game Crysis running on their new engine, the CryEngine 2, and it was amazing. The graphical superiority was evident as it could render massive open worlds, filled with vegetation. Crysis itself was an award winning game but also needed a very powerful computer to run it. This spawned the popular saying, "Can it play Crysis?", when discussing computer power. The game Crysis is still used nowadays to benchmark computer specifications. The CryEngine 2 greatly improved upon its predecessor in every way, adding a variety of new technologies as well as tools available for modders and developers. The game Crysis came with a Sandbox editor which was used to make mods and add features. The great feature of this editor was that after the player had finished tweaking things or placing items, with the touch of a button they could instantly drop in as the player character with no load times. Crytek named this the "What You See Is What You Play" editor style. Some of the main features in the CryEngine were Polybump 2 and its physics system. Its physics system is different as it is developed in house, which is different than the Unreal 3 engines as they used the Nvidia PhysX software, and also allowed for liquid physics. Usually liquids in games would be static, they would move as the water textures moved, or display ripple textures when players walked in them, but the CryEngine was able to render the liquids as something physical which could be interacted with. This technique greatly added to the realness of the game and the engines capabilities, but took a lot of power from the machine to implement. Polybump 2 is an in house technology developed by Crytek which "creates a high quality surface description that allows quick access of surface features like normal maps, displacement maps, unoccluded area direction, and accessibility"¹⁸. This feature allowed

¹⁷ <http://www.bbc.co.uk/news/technology-17535906>

¹⁸ <http://www.crytek.com/cryengine/cryengine2/overview>

greater than normal detailing but using less processing power than usual. As well as being licenced out to game developers CryEngine 2 was licenced to French company IMAGTP, a company who specialise in architecture and urban planning, to make simulations of what their buildings would look like before a brick was laid. I could not find a website but here is a video showing one of their simulations¹⁹.

A Newcomer and an old Master

2011 was a very busy year for game engines. The Unreal 3 engine added Mac OS support and the Source engine added Playstation 3 support, both greatly expanding their market. It also saw fully new engines from Crytek, DICE, and id software.

DICE is famous for making the battlefield series of games, which are all first person shooter games, set on massive maps and implementing full vehicle combat. It has always made its own engines and in 2008 it released Battlefield: Bad Company which debuted its Frostbite engine. The defining feature in the engine was its destructible terrain, which made the vehicle combat very visceral and real. In 2009 DICE released version 1.5 of the engine with a better destruction model named Destruction 2.0. DICE never licenced this engine and only used it themselves for their own games. In 2011 DICE released Battlefield 3 along with their new engine Frostbite 2. DICE are owned by EA and since releasing Frostbite 2 have licenced the engine to several other EA studios, id software are doing the same thing with their new engine. The engine itself has won numerous awards and DICE have done presentations at SIGGRAPH as well as GDC demonstrating the new features of their engine²⁰. The new engine has increased destruction capabilities, realistic lighting, and a ground breaking animation system.

Id software also released a new game and engine in 2011, their first since DOOM 3 in 2004. The game was called Rage and the engine was dubbed id tech 5, instead of just the Rage engine. When this engine was released all previous id engines were renamed 'id tech X' instead of the game they were made for. Rage was the first game id made that was released on the Xbox 360 and the Playstation 3, having only developed for the PC before. Before releasing Rage id signed with publisher Bethesda Softworks, so their game could be published on consoles. When id released the engine they stated that they would no longer be

¹⁹ <http://www.youtube.com/watch?v=B6spFgJWC1c>

²⁰ <http://publications.dice.se/>

licencing their engine to developers outside Bethesda²¹. That meant that any developer wanted to use id tech 5 would have to sign a publishing deal with Bethesda to licence it. The engine itself is vastly superior to the DOOM 3 engine, or id tech 4. One of the major changes in the engine was that it was designed to benefit developers and make their job easier, whereas old id engines generally focused on the player rather than developers. It also greatly improved on the MegaTexture technique, now called Virtual Texturing, from id tech 4. The new engine could map textures up to the size of 128,000 x 128,000 pixels and not only be applied to landscapes but items and characters as well.

Crytek also released Crysis 2 in 2011 showcasing their new CryEngine 3 engine. The new engine was built upon the old CryEngine with many improvements. CryEngine 2 had some of the best graphics at its time of release, and CryEngine 3 improved on these, but the main area it improved upon was its scalability. It could change more game aspects than usual depending on how powerful the machine was that it was being played on. It was still demanding for machines playing it, but not nearly as synonymous as its predecessor. An example of this was Crytek re-released the original Crysis built using CryEngine 3 for the Xbox 360. The hardware requirements for the Xbox 360 could not support the original on CryEngine 2, but the improved engine achieved this with ease. A few months after Crysis 2 was released, Crytek released the CryEngine 3 SDK for free, non-commercially. This was probably in response to Epic releasing their UDK, as UDK turned out to be very popular and widely used around the world. It could be used for developing games and simulations, and these could be distributed for free without a licence, but commercial distribution requires a licence. Unlike the UDK it does not publicly state this indie licence terms. CryEngine 3 has been the most widely licenced engine from Crytek, both in the gaming realm and in simulations. So much so that Crytek set up another studio to deal specifically with simulation based licences for companies not involved in creating games. I think this shows how powerful the CryEngine is and life like the scenes it can make are. One of its biggest licences was to the US army with a massive price of fifty seven million dollars²².

²¹ <http://www.joystiq.com/2010/08/12/id-tech-5-exclusive-to-bethesda-published-titles/>

²² <http://gaming.operationreality.org/2011/05/28/us-army-goes-with-cryengine3-military-simulator/>

Conclusion

Every iteration of a game engine introduces new and amazing technology. The power they can achieve, as evidenced by simulation licences, is well beyond the scope of just games at this stage, and future game engines will push the boundaries even more.

John Carmack has been talking for a long time about the technology he would like to see in id tech 6. He plans on mixing the standard way of creating graphics, rasterization, with ray tracing²³, but said during a keynote speech at QuakeCon 08 that the hardware capable of this does not exist yet.

The Middleware for Middleware

I decided to add this part in at the end to separate it from the full game engine discussion. As stated earlier game engines are middleware, but developers have created their own middleware to be used in game engines for a long time. The purpose of this middleware varies greatly, and they are more specialised than game engines. Here I will talk about the four biggest middleware applications available for game engines.

By far the most widely used middleware application is Bink Video²⁴. Bink Video is a video codec for games that scales its data rate depending on what platform it is being used on, it also is compatible with fourteen different platforms, giving it great usability. On its own site it states “it has been licensed for over 5,800 games”, which is more than any game engine has ever gotten close to.

Another widely used middleware application is called Scaleform²⁵. Scaleform is a flash based application that creates the user interface in games, such as main menus and it has been used in over one thousand games. Scaleform and Bink showcase how a certain niche application can achieve such amazing number of licenses from doing their given job extremely well and better than anyone else.

Another middleware application that occupies this niche area is SpeedTree. SpeedTree is a toolkit used to create vegetation for games, as well as films. It allows developers to quickly create vegetation that would be very tome consuming if they were to

²³ <http://www.pcper.com/reviews/Graphics-Cards/John-Carmack-id-Tech-6-Ray-Tracing-Consoles-Physics-and-more>

²⁴ <http://www.radgametools.com/bnkmain.htm>

²⁵ <http://gameware.autodesk.com/scaleform>

program and create them from scratch. It also can be used to self-populate a game world with trees so a developer doesn't even have to place them. One of the most notable films SpeedTree has been used for is Avatar by James Cameron.

The last middleware application I will talk about is Havok software. Havok software was started by two ex-computer science students of Trinity College Dublin and has become one of the most used physics engines worldwide. Havok started out as just the Havok Physics engine but more features were added to it over time, such as, Havok Cloth, which simulates cloths and soft body dynamics, and Havok Destruction, which simulates destruction. It has been used in over one hundred and sixty games and has also branched out to movies. Some of the most famous movies to use Havok are The Matrix, Watchmen, X-Men: The Last Stand, and Harry Potter and the Deathly Hallows.