RESEARCH ARTICLE OPEN ACCESS

An Overview Study of Game Engines

Faizi Noor Ahmad
Student at Department of Computer Science, ACNCEMS (Mahamaya Technical University), Aligarh-202002, U.P., India

ABSTRACT
We live in a world where people always try to find a way to escape the bitter realities of hubbub life. This escapism gives rise to indulgences. Products of such indulgence are the video games people play. Back in the past the term "game engine" did not exist. Back then, video games were considered by most adults to be nothing more than toys, and the software that made them tick was highly specialized to both the game and the hardware on which it ran. Today, video game industry is a multi-billion-dollar industry rivaling even the Hollywood. The software that drives these three dimensional worlds- the *game engines*-have become fully reusable software development kits. In this paper, I discuss the specifications of some of the top contenders in video game engines employed in the market today. I also try to compare up to some extent these engines and take a look at the games in which they are used.
Keywords **–** engines comparison, engines overview, engines specification, video games, video game engines

I. INTRODUCTION

Back in the past the term "game engine" did not exist. Back then, video games were considered by most adults to be nothing more than toys, and the software that made them tick was highly specialized to both the game and the hardware on which it ran. Today, video game industry is a multi-billion-dollar industry rivalling even the Hollywood. The software that drives these three dimensional worlds- the *game engines*-have become fully reusable software development kits. Virtually all game engines contain a familiar set of core components, including the rendering engine, the collision and physics engine, the animation system, the audio system, the game world object model, the artificial intelligence system, and so on.

II. STRUCTURE OF A GAME TEAM[1]

Let's first take a brief look at the structure of a typical game development team. Game studios are usually composed of five basic disciplines: engineers, artists, game designers, producers, and other management and support staff (marketing, legal, information technology/technical support, administrative, etc.). Each discipline can be divided into various sub-disciplines. We'll take a brief look at each

1.1.1 Engineers

The engineers design and implement the soft ware that makes the game, and the tools, work. Engineers are oft en categorized into two basic groups: *runtime* programmers (who work on the engine and the game itself) and *tools* programmers (who work on the off -line tools that allow the rest of the development team to work effectively).

1.1.2 Artists

The artists produce all of the visual and audio content in the game, and the quality of their work can literally make or break a game.

- *Concept artists* produce sketches and paintings that provide the team with a vision of what the final game will look like. They start their work early in the concept phase of development, but usually continue to provide visual direction throughout a project's life cycle. It is common for screen shots taken from a shipping game to bear an uncanny resemblance to the concept art.
- *3D modellers* produce the three-dimensional geometry for everything in the virtual game world. This discipline is typically divided into two sub disciplines: foreground modellers and background modellers. The former create objects, characters, vehicles, weapons, and the other objects that populate the game world, while the latter build the world's static background geometry (terrain, buildings, bridges, etc.).
- *Texture artists* create the two-dimensional images known as textures, which are applied to the surfaces of 3D models in order to provide detail and realism.
- *Animators* imbue the characters and objects in the game with motion. The animators serve quite literally as actors in a game production, just as they do in a CG film production. However, a game animator must have a unique set of skills in order to produce animations that mesh seamlessly with the technological underpinnings of the game engine.
- *Motion capture actors* are often used to provide a rough set of motion data, which are then cleaned

up and tweaked by the animators before being integrated into the game.
- *Sound designers* work closely with the engineers in order to produce and mix the sound effects and music in the game.
- *Voice actors* provide the voices of the characters in many games.

1.1.3 Game Designers

The game designers' job is to design the interactive portion of the player's experience, typically known as *gameplay*. Different kinds of designers work at different levels of detail. Some (usually senior) game designers work at the macro level, determining the story arc, the overall sequence of chapters or levels, and the high-level goals and objectives of the player. Other designers work on individual levels or geographical areas within the virtual game world, laying out the static background geometry, determining where and when enemies will emerge, placing supplies like weapons and health packs, designing puzzle elements, and so on. Still other designers operate at a highly technical level, working closely with gameplay engineers and/or writing code (often in a high-level scripting language). Some game designers are ex-engineers, who decided they wanted to play a more active role in determining how the game will play.

1.1.4 Producers

The role of producer is defined differently by different studios. In some game companies, the producer's job is to manage the schedule and serve as a human resources manager. In other companies, producers serve in a senior game design capacity. Still other studios ask their producers to serve as liaisons between the development team and the business unit of the company (finance, legal, marketing, etc.). Some smaller studios don't have producers at all. For example, at Naughty Dog, literally everyone in the company, including the co-presidents, play a direct role in constructing the game; team management and business duties are shared between the senior members of the studio.

1.1.5 Publishers

The marketing, manufacture, and distribution of a game title are usually handled by a *publisher*, not by the game studio itself. A publisher is typically a large corporation, like Electronic Arts, THQ, Vivendi, Sony, Nintendo, etc. Many game studios are not affiliated with a particular publisher. They sell each game that they produce to whichever publisher strikes the best deal with them. Other studios work exclusively with a single publisher, either via a long term publishing contract, or as a fully owned subsidiary of the publishing company. For example, THQ's game studios are independently managed, but they are owned and ultimately controlled by THQ. Electronic Arts takes this relationship one step further, by directly managing its studios. *First-party developers* are game studios owned directly by the console manufacturers (Sony, Nintendo, and Microsoft). For example, Naughty Dog is a first-party Sony developer. These studios produce games exclusively for the gaming hardware manufactured by their parent company.[2]

III. WHAT IS A GAME?

The word "game" usually conjures images of a three-dimensional virtual world featuring a humanoid, animal, or vehicle as the main character under player control. (Or for the old ones among us, perhaps it brings to mind images of two-dimensional classics like Pong, Pac-Man, or Donkey Kong.) Raph Koster defines a "game" to be an interactive experience that provides the player with an increasingly challenging sequence of patterns which he or she learns and eventually masters. Koster's assertion is that the activities of learning and mastering are at the heart of what we call "fun," just as a joke becomes funny at the moment we "get it" by recognizing the pattern. Specifically, a video game is a game played by electronically manipulating images produced by a computer program on a television screen or display.

IV. WHAT IS A GAME ENGINE

The term "game engine" arose in the mid-1990s in reference to first-person shooter (FPS) games like the insanely popular *Doom* by id Soft ware. *Doom* was architected with a reasonably well-defined separation between its core software components (such as the three-dimensional graphics rendering system, the collision detection system, or the audio system) and the art assets, game worlds, and rules of play that comprised the player's gaming experience. The value of this separation became evident as developers began licensing games and re-tooling them into new products by creating new art, world layouts, weapons, characters, vehicles, and game rules with only minimal changes to the "engine" software. This marked the birth of the "mod community"—a group of individual gamers and small independent studios that built new games by modifying existing games, using free toolkits provided by the original developers. Towards the end of the 1990s, some games like *Quake III Arena* and *Unreal* were designed with reuse and "modding" in mind. Engines were made highly customizable via scripting languages like id's Quake C, and engine licensing began to be a viable secondary revenue stream for the developers who created them.

The line between a game and its engine is oft en blurry. Some engines make a reasonably clear distinction, while others make almost no attempt to separate the two. In one game, the rendering code might "know" specifically how to draw an orc. In another game, the rendering engine might provide general-purpose material and shading facilities, and "orc-ness" might be defined entirely in data. No studio

makes a perfectly clear separation between the game and the engine, which is understandable considering that the definitions of these two components oft en shift as the game's design solidifies. Arguably a *data-driven architecture* is what differentiates a game engine from a piece of software that is a game but not an engine. When a game contains hard-coded logic or game rules, or employs special-case code to render specific types of game objects, it becomes difficult or impossible to reuse that software to make a different game. We should probably
reserve the term "game engine" for software that is extensible and can be used as the foundation for many different games without major modification.[3]

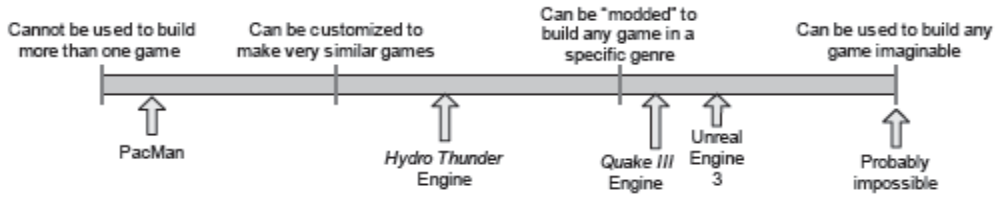

Figure 1.1. Game engine reusability gamut.

A Game Engine provides the core functionalities of a game:

- It manages the objects, game levels, and other resources in the game world
- It renders the viewable portion of the world to the user
- It handles updating of the positions of objects
- It also manages the behaviour of non-player characters

V. ENGINES ACROSS GENRES

Game engines are typically somewhat genre specific. An engine designed for a two-person fighting game in a boxing ring will be very different from a massively multiplayer online game (MMOG) engine or a first-person shooter (FPS) engine or a real-time strategy (RTS) engine. However, there is also a great deal of overlap—all 3D games, regardless of genre, require some form of low-level user input from the joypad, keyboard, and/or mouse, some form of 3D mesh rendering, some form of heads-up display (HUD) including text rendering in a variety of fonts, a powerful audio system, and the list goes on. So while the Unreal Engine, for example, was designed for first-person shooter games, it has been used successfully to construct games in a number of other genres as well, including the wildly popular third-person shooter *Gears of War* by Epic Games; the character-based action-adventure game *Grimm*, by American McGee's Shanghai-based development studio, Spicy Horse; and *Speed Star*, a futuristic racing game by South Korea-based Acro Games[4].

1.4.1 First Person Shooters

First-person shooter (FPS) is a video game genre centred on gun and projectile weapon-based combat through a first-person perspective; that is, the player experiences the action through the eyes of the protagonist. The first-person shooter (FPS) genre is typified by games like *Quake, Unreal Tournament, Half-Life, Counter-Strike, Crysis, Farcry* and *Call of Duty*. These games have historically involved relatively slow on-foot roaming of a potentially large but primarily corridor-based world. However, modern first-person shooters can take place in a wide variety of virtual environments including vast open outdoor areas and confined indoor areas. Modern FPS traversal mechanics can include on-foot locomotion, rail-confined or free-roaming ground vehicles, hovercraft , boats, and aircraft. First-person games are typically some of the most technologically challenging to build, probably rivaled in complexity only by third-person shooter/action/platformer games and massively multiplayer games. This is because first-person shooters aim to provide their players with the illusion of being
immersed in a detailed, hyper realistic world. It is not surprising that many of the game industry's big technological innovations arose out of the games in this genre.
First-person shooters typically focus on technologies, such as

- efficient rendering of large 3D virtual worlds;
- a responsive camera control/aiming mechanic;
- high-fidelity animations of the player's virtual arms and weapons;
- a wide range of powerful hand-held weaponry;
- a forgiving player character motion and collision model, which oft en gives these games a "floaty" feel;
- high-fidelity animations and artificial intelligence for the non-player characters (the player's enemies and allies);
- small-scale online multiplayer capabilities (typically supporting up to 64 simultaneous players), and the ubiquitous "death match" gameplay mode.

A first-person shooter engine is a video game engine specialized for simulating 3D environments for use in a first-person shooter video game. First-person

refers to the view where the players see the world from the eyes of their characters. Shooter refers to games which revolve primarily around wielding firearms and killing other entities in the game world, either NPCs or other players. The development of the FPS graphic engines is characterized by a steady increase in technologies, with some breakthroughs. Attempts at defining distinct generations lead to arbitrary choices of what constitutes a highly modified version of an 'old engine' and what is a brand new engine.The classification is complicated as game engines blend old and new technologies. Features considered advanced in a new game one year, become the expected standard the next year. Games with a combination of both older and newer features are the norm. For example *Jurassic Park: Trespasser* (1998) introduced physics to the FPS genre, which didn't become common until around 2002. *Red Faction* (2001) featured destructible walls and ground, something still not common in engines years later.

1.4.1.1 Some FPS video games and their engines

- Call of Duty series

Call of Duty is a first-person and third-person shooter computer/video game franchise. The series began on the PC, and later expanded to consoles and handhelds. Several spin-off games have also been released. The earlier games in the series are set primarily in World War II, including Call of Duty, Call of Duty 2, and Call of Duty 3. Beginning with Modern Warfare, which is set in modern times, the series has shifted focus away from World War II. Call of Duty 4: Modern Warfare (released November 2007) was followed by Call of Duty: World at War and Call of Duty: Modern Warfare 2. Black Ops (released November 2010) takes place in the Cold War, while Modern Warfare 3 (released November 2011) takes place in the near-future setting. Black Ops 2 (released November 2012) takes place in the year 2025. The game uses a variety of game engines but the one used recently is IW (infinity ward) engine in modern warfare 3 and a modified version in black ops II.

- Crysis

Crysis is a first-person shooter video game series developed by German developer Crytek and published by Electronic Arts. The series revolves around a group of military protagonists with "nanosuits," technologically advanced suits of armor that allow them to gain enhanced physical strength, speed, defense, and cloaking abilities. The protagonists face off against hostile North Korean soldiers, heavily-armed mercenaries, and a race of technologically advanced aliens known as the Ceph, who arrived on Earth millions of years ago for unclear reasons, and have recently been awakened. As of 2013, the series consists of three main installments, a standalone spinoff of the first game with a separate multiplayer title, and a compilation. It uses a game engine called CryEngine which is modified with time finally for the most recent instalment Crysis 3.

1.4.2 Third Person games

Third-person character-based games have a lot in common with first-person shooters, but a great deal more emphasis is placed on the main character's abilities and locomotion modes. In addition, high-fidelity full-body character animations are required for the player's avatar, as opposed to the somewhat less-taxing animation requirements of the "floating arms" in a typical FPS game. It's important to note here that almost all first-person shooters have an online multiplayer component, so a full-body player avatar must be rendered in addition to the first-person arms. Third-person is a perspective in which the player can visibly see the body of the controlled character. This is seen in most third-person shooters and adventure games. It most commonly refers to a viewpoint behind the player character. Third-person perspective puts you in direct control of a character that you can actually see in front of you. Third-person, in grammar is referred to as he, she or it. This means that you are not directly the person you are controlling. You do not control his thoughts or what he looks at (the first-person perspective, the view, gives you better control over putting you into someone else's body even if, in a grammar sense, it does not really count as first-person). The third-person perspective usually has the player looking at his character's back for most of the game. The character is usually on the the left side of the screen and the camera angle may be fixed behind the character meaning you never see his face. Some games have the character in the centre of the screen where you can pan the camera around the character (Uncharted or GTA). If the camera is fixed, than panning the camera controls the character's view. Aiming (in shooters) is usually done over-the-shoulder. The camera zooms over the shoulder of the character, where the player can aim. Older games (the behind-the-shoulder trend was started by Resident Evil 4) have you just shooting at enemies. Since this can be imprecise, some third-person games have a lock on targeting system. In order to immerse the player, some third person games use different techniques to make the player feel like he's watching a movie. Blood spatters onto the screen in Gears of War, shaky cam is used during intense moments in Uncharted, lens flare appears in Blood Stone when the player is facing a light and what is maybe the most original use of the third person perspective to this date, Dog Days makes it look like a camera man is following you around.

Some of the technologies specifically focused on by games in this genre include

- moving platforms, ladders, ropes, trellises, and other interesting locomotion modes;
- puzzle-like environmental elements;

- a third-person "follow camera " which stays focused on the player character and whose rotation is typically controlled by the human player via the right joypad stick (on a console) or the mouse (on a PC—note that while there are a number of popular third-person shooters on PC, the platformer genre exists almost exclusively on consoles);
- a complex camera collision system for ensuring that the view point never "clips" through background geometry or dynamic foreground objects.

1.4.2.1 Some TPS games

- Grand Theft Auto

Grand Theft Auto, officially abbreviated to GTA, is a video game series created by game programmers David Jones and Mike Dailly then later by brothers Dan and Sam Houser, and game designer Zachary Clarke. It is primarily developed by Rockstar North (formerly DMA Design) in the United Kingdom and published by Rockstar Games. The name of the series is derived from "grand theft auto", a term referring to motor vehicle theft. The series has won multiple awards, and is named as one of the most universally acclaimed gaming franchises, with Grand Theft Auto III and IVconsidered as two of the most revolutionary and critically acclaimed video games of all time. The series is set in fictional locales heavily modelled on American cities, while an expansion for the original was based in London. Gameplay focuses on an open world where the player can choose missions to progress an overall story, as well as engaging in side activities, all consisting of action-adventure, driving, occasional role-playing, stealth, and racing elements. The subject of the games is usually a comedic satire of American culture, but the series has gained controversy for its adult nature and violent themes. The series focuses around many different protagonists who attempt to rise through the ranks of the criminal underworld, although their motives for doing so vary in each game. The antagonists are commonly characters who have betrayed the protagonist or his organisation, or characters who have the most impact impeding the protagonist's progress. The game uses Rockstar's own RAGE game engine, which was previously used in Rockstar Table Tennis, in combination with the Euphoria game animation engine. Instead of pre-written animations, Euphoria uses procedural animation to control the way the player moves, enabling character movements to be more realistic. The Euphoria engine also enables NPCs to react in a realistic way to the player's actions. In one preview, a player knocked an NPC out of a window and the character grabbed onto a ledge to stop himself from falling. The game also uses middleware from Image Metrics to facilitate intricate facial expressions and ease the process of incorporating lip-synching. Foliage in the game is produced through SpeedTree.

- Assassins Creed

Assassin's Creed is a historical fiction action-adventure open world stealth video game series that consists of five main games and a number of supporting materials, as of 2013. The games have appeared on the PlayStation 3, Xbox 360, Microsoft Windows, Mac OS X, Nintendo DS, PlayStation Portable, PlayStation Vita, iOS, HP webOS, Android, Nokia Symbian, Windows Phone platforms, and the Wii U. The main games in the franchise were developed by Ubisoft Montreal for the single player and Ubisoft Annecy for the multiplayer, with the handheld titles developed by Gameloft and Gryptonite Studios, with additional development by Ubisoft Montreal. The series has been well received by the public and critics, and has sold over 55 million copies as of March, 2013. While playing as the Assassin characters, the games are generally presented as third-person in an open world, focusing on stealth and free-running. The games use a mission structure to follow the main story, generally assigning the player to complete an assassination of public figureheads or a covert mission. Alternatively, several side missions are available, such as mapping out the expansive cities from a high perch followed by performing a "leap of faith" into a haystack below, collecting treasures hidden across the cities, exploring ruins for relics, building a brotherhood of assassins to perform other tasks, or funding the rebuilding of a city through purchasing and upgrading of shops and other features. At times, the player is in direct control of Desmond, who by nature of the Animus use has learned Assassin techniques through the bleeding effect, as well as their genetic ability of Eagle Vision, which separates friend, foe and assassination targets by illuminating people in different colours. Through the Animus interface, the player can go back to retry any past mission already completed; for example, in Assassin's Creed: Brotherhood, the player achieves better synchronization results by performing the mission in a specific manner such as by only killing the mission's target. The games use the concept of "active" versus "passive" moves, with "active" moves, such as running, climbing the sides of buildings, or jumping between rooftops, more likely to alert the attention of nearby guards. When the guards become alerted, the player must either fight them or break their line of sight and locate a hiding place, such as a haystack or a well, and wait until the guards' alert is reduced. The combat system allows for a number of unique weapons, armour, and moves, including the use of a hidden blade set in a bracer on the Assassin's arm, and which also can be used to quietly assassinate targets. The first game in the series uses an engine by name of 'scimitar' which modifies to be called 'anvil' and 'anvilnext' used in the most recent instalment i.e. Assassins Creed III. AnvilNext, delivers improved

visuals, character models and AI, allowing for battlefields full of fighters. Ubisoft later said it's aiming to make Assassin's Creed III "look next-gen" on current-gen consoles using the new AnvilNext engine.

1.4.3 Racing Games

The racing genre encompasses all games whose primary task is driving a car or other vehicle on some kind of track. The genre has many subcategories. Simulation-focused racing games ("sims") aim to provide a driving experience that is as realistic as possible (e.g., Gran Turismo). Arcade racers favour over-the-top fun over realism (e.g., San Francisco Rush, Cruisin' USA, Hydro Thunder). A relatively new subgenre explores the subculture of street racing with tricked out consumer vehicles (e.g., Need for Speed, Juiced). Kart racing is a subcategory in which popular characters from platformer games or cartoon characters from TV are re-cast as the drivers of whacky vehicles (e.g., Mario Kart, Jak X, Freaky Flyers). "Racing" games need not always involve time-based competition. Some kart racing games, for example, offer modes in which players shoot at one another, collect loot, or engage in a variety of other timed and untimed tasks.

Some of the technological properties of a typical racing game include the following techniques.

- Various "tricks" are used when rendering distant background elements, such as employing two-dimensional cards for trees, hills, and mountains.
- The track is oft en broken down into relatively simple two-dimensional regions called "sectors." These data structures are used to optimize rendering and visibility determination, to aid in artificial intelligence and path finding for non-human-controlled vehicles, and to solve many other technical problems.
- The camera typically follows behind the vehicle for a third-person perspective, or is sometimes situated inside the cockpit first-person style.
- When the track involves tunnels and other "tight" spaces, a good deal of effort is oft en put into ensuring that the camera does not collide with background geometry.

1.4.3.1 Some Racing video games

- Need for Speed (NFS)

Undoubtedly, most famous video game series in racing genre Need for Speed (NFS) is a series of racing video games published by Electronic Arts (EA) and developed by several studios including Canadian company EA Black Box and British company Criterion Games. The series released its first title, The Need for Speed in 1994. Initially, Need for Speed was exclusive to the fifth generation video game consoles, eventually featuring in all seventh generation consoles by 2008. The titles consist of racing with different cars on various tracks, with some titles including police pursuits in races. Since Need for Speed: Underground, the series has integrated car body customization into gameplay. Need for Speed is the most successful racing video game series in the world, and one of the most successful video game franchises of all time. As of October 2009, over 140 million copies of games in the Need for Speed series have been sold. The most recent instalment in this series is Most Wanted and The Run. The Run is powered by DICE's Frostbite 2 engine, making the game the first non-shooter and one of the first console titles to use the engine, which provides visuals and car physics that "hug the road even at top speeds all built around a gripping storyline." Additionally, Need for Speed Autolog, the Need for Speed franchise's social competition functionality, which was introduced in Hot Pursuit and was previously used in Shift 2: Unleashed, is also back as it continues to track career progression and compare game stats.

- Dirt

Colin McRae Rally and more recently; Dirt, is a racing video game series developed and published by Codemasters. Started in 1998, the franchise has been a critical and commercial success and is generally acknowledged as a pioneer of realistic rally sportsracing games. The series is named after the World Rally Championship driver Colin McRae, who provided technical advice during development. The most recent games use ego 2.0 game engine. Ego Game Technology Engine (more commonly referred to as Ego Engine or EGO, stylised ego) is a video game engine developed by Codemasters. Ego is a modified version of the Neon game engine that was used in Colin McRae: Dirt and was developed by Codemasters and Sony Computer Entertainment using Sony Computer Entertainment's Phyre Engine cross-platform graphics engine. The Ego engine was developed to render more detailed damage and physics as well as render large-scale environments[5].

VI. GAME ENGINE SURVEY[6]

1.5.1 RAGE

Company: Rockstar

As Seen In: Max Payne 3, L.A. Noire, GTA IV, Midnight Club: Los Angeles, Red Dead Redemption

Rockstar Advanced Game Engine (RAGE) is a game engine developed by the RAGE Technology Group at the video game developer Rockstar San Diego with contributions by other Rockstar studios. Rockstar developed the engine to facilitate game development on Microsoft Windows, along with the PlayStation 3, Wii, and Xbox 360 consoles. RAGE evolved from the Angel Game Engine originally developed by Angel Studios for use in the sixth generation console era versions of the Midnight Club series and other Rockstar San Diego games. Rockstar has integrated a few third party middleware components

into RAGE like the proprietary Euphoria character animation engine and the open source Bullet physics engine. Prior to RAGE, Rockstar mostly used Criterion Games' RenderWare engine to develop various game titles, including the PlayStation 2, Xbox, and Microsoft Windows instalments of the Grand Theft Auto franchise. Since the release of Max Payne 3, the engine supports DirectX 11 and stereoscopic 3D rendering on the PC platform.

The RAGE engine consists of a collection of core game technologies. Its basis is the AGE engine that was developed for previous Rockstar San Diego titles. It contains a rendering framework, a physics engine, audio engine, network libraries, animation engine, scripting language plus loads more. It is mainly developed at Rockstar San Diego apart from the audio engine and a lot of the network code which were developed here at North.

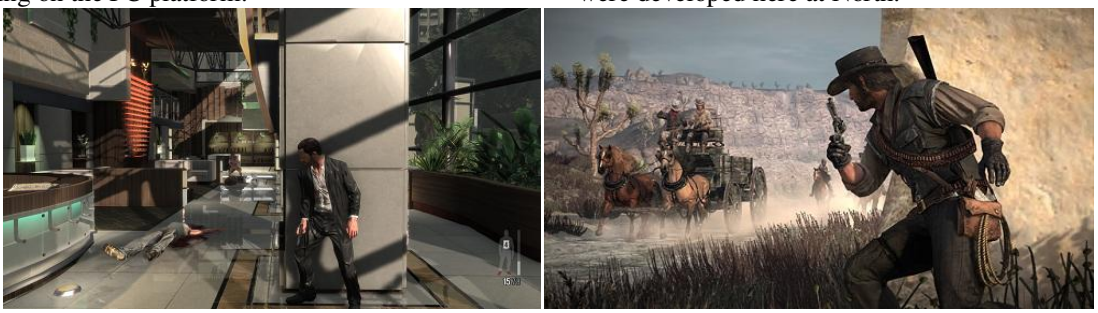

Fig 1.2 Lighting and environment effects of RAGE in Max Payne 3(L) and Red Dead Redemption(R)

1.5.2 CryEngine
Company: Crytek
As seen in: Far Cry, Crysis, Crysis Warhead, Crysis 2, Crysis 3

Crytek's in-house engine has been appearing in a bevy of games lately, not least the developer's own Crysis 3, Warface and Ryse. Suitable for PC, PS3 and Xbox 360, CryEngine 3 is capable of generating state-of-the-art graphics, lighting, physics and AI, as well as having a custom-made sandbox editor. A free version is also available for non commercial use via its website.

Features include:

- Simultaneous WYSIWYG on all platforms in Sandbox editor
- "Hot-update" for all platforms in Sandbox editor
- Material Editor
- Flow graph
- Track View Editor
- Procedural Placement Tools & Cover Generation
- Integrated vegetation & terrain cover generation system
- Real time soft particle system & integrated FX editor
- Road & river tools
- Vehicle creator
- Fully Flexible Time of Day System
- Streaming
- Performance Analysis Tools
- Visual Budget System
- Multi-core support
- Sandbox development layers
- Offline rendering
- Resource compiler
- Natural lighting & dynamic soft shadows with penumbra
- Irradiance Volume
- Deferred lighting
- Real time dynamic global illumination
- Eye adaptation & high dynamic range(HDR) lighting
- Screen Space Ambient Occlusion
- Colour grading
- "Uber Shader" technology
- Blend Layer
- Normal maps & parallax occlusion maps
- Motion blur & depth of field with sprite-based bokeh
- High quality 3D water
- Dynamic volumetric light beams & light shaft effects
- Volumetric, layer & view distance fogging etc.

1.5.2.1 Versions

- CryEngine 1

CryEngine is a game engine used for the first-person shooter video game Far Cry. It was originally developed by Crytek as a technology demo for Nvidia and, when the company saw its potential, it was turned into a game. When video cards with support for 3.0 pixel and vertex shaders were released, Crytek released version 1.2 of the engine which used some of the capabilities for better graphics. Later the company developed CryEngine version 1.3, which added support for HDR lighting. The engine has been licensed to NCsoft for their MMORPG, Aion: The Tower of Eternity. On March 30, 2006, Ubisoft acquired all intellectual property rights to the Far Cry franchise and a perpetual license to use the Far Cry edition of CryEngine.

- CryEngine2

CryEngine 2 is used in Crytek's game Crysis, and an updated version in Crysis Warhead, a side story of Crysis. In March 2009 at the Game

Developers Conference, CryEngine 2's successor, Cry Engine 3, was shown on the Xbox 360 and Play Station 3.CryEngine 2 was first licensed out to French company IMAGTP who specializes in architectural and urban-planning communication. The purpose of licensing the engine was to create a program to allow clients to see exactly what a building or other structure would look like before any actual building was undertaken. As of March 7, 2011, Simpson Studios has licensed CryEngine 2 out to use on a Massively Multiplayer Virtual World (MMVW) that takes place on a terraformed Mars. On May 11, 2007, Crytek announced that they would be using the engine to create a game based on their new "intellectual property". It is also confirmed that it will not be a part of Crysis and in fact may not even be a first person shooter. On September 17, 2007, Ringling College of Art & Design became the first higher education institution in the world to license CryEngine 2 for educational purposes.

- CryEngine 3

On March 11, 2009, the German game studio Crytek announced that it would introduce CryEngine 3 at the 2009 Game Developers Conference, held from March 25 to March 27. The new engine was being developed for use on Microsoft Windows, PlayStation 3, Xbox 360, and Wii U. As for the PC platform, the engine is said to support development in DirectX 9, 10, and 11. As of June 1, 2009, it was announced that Crysis 2 would be developed by Crytek on their brand new engine. CryEngine 3 was released on October 14, 2009. On March 1, 2010, a new tech demo of the engine was released for the i3D 2010 symposium, which demonstrates 'Cascaded Light Propagation Volumes for Real Time Indirect Illumination'. On June 11, 2011, the Australian Defence Force revealed that Navy personnel would train on a virtual landing helicopter dock ship made using the CryEngine 3 software. As of July 1, 2011, the Mod SDK version of CryEngine 3 specifically to create custom maps, mods and content for Crysis 2 is available on Crytek's website. Crytek also released a free-to-use version of the CryEngine for non-commercial game development. It was released as of August 17, 2011 under the name CryEngine 3 SDK. Crytek announced on September 9, 2011 that they would be using CryEngine 3 to bring the original Crysis to consoles. It was released for Xbox Live and PlayStation Network on October 4, 2011.

CryEngine 3 Free SDK, originally called Sandbox Editor, is the current version of the level editor used to create levels for the CryEngine line of game engines by Crytek. Tools are also provided within the software to facilitate scripting, animation, and object creation. It has been included with various Crytek games (including, but not limited to, Crysis and Far Cry), and is used extensively for modding purposes. The editing style is that of the sandbox concept, with the emphasis on large terrains and a free style of mission programming. The editor can also construct indoor settings. As opposed to editors like UnrealEd, which use a "subtractive" editing style that takes away areas from a filled world space, the Sandbox has an "additive" style (like Quake II). Objects are added to an overall empty space.The Sandbox's concentration on potentially huge (in theory, hundreds of square kilometres) terrain, means that it uses an algorithmic form of painting textures and objects onto the landscape. This uses various parameters to define the distribution of textures or types of vegetation. This is intended to save time and make the editing of such large terrains feasible while maintaining the overall "real world" sandbox free roaming style. This is different from some editing styles that often use "fake backdrops" to give the illusion of large terrains. In a fashion somewhat comparable to the 3D Renderer Blender, which can be used for game design, the Sandbox editor has the ability, with a single key press, for the editor to jump straight into the current design (WYSIWYP, "What You See Is What You Play" Feature). This is facilitated without loading the game as the game engine is already running within the editor. The "player" view is shown within the 3D portion of the Editor. The Editor also supports all the CryEngine features such as vehicles and physics, scripting, advanced lighting (including real time, moving shadows), Polybump technology, shaders, 3D audio, character Inverse kinematics and animation blending, dynamic music, Real Time Soft Particle System and Integrated FX Editor, Deferred Lighting, Normal Maps & Parallax Occlusion Maps, and Advanced Modular AI System.

Table 1 and Table 2 shows the comparison of CryEngine with some other engines in terms of certain features.

Game Engines	Platform	Language Support	API	Physics Engine	Path finding
CryEngine	Win,X360, PS3	C++, Visual Script	DX9, 10,11	Custom	Yes
Hero Engine	Win	C++, Hero Script	DX 9	Physx	Yes
Source Engine	Win, Mac, X360, PS3	C++	DX9, Open GL	Custom	Yes
Unity	Win, Mac, Android, X360, PS3, Wii	C#, Java Script	DX9, Open GL	Physx	No
Unreal Engine 3	Win, X360, PS3	C++, Unreal Script	DX9, 10, 11	Physx	Yes
Vision Engine	Win, X360, PS3, Wii	C++	DX9, 10, 11	Physx, Bullet	Yes

Table 1

Engines	World Editor	GUI Editor	Deferred renderer	Forward renderer	Built-in MMO
Cry Engine	Yes	No	Yes	No	No
Hero Engine	Yes	No	No	Yes	Yes
Source Engine	Yes	No	No	Yes	No
Unity	Yes	No	Yes	Yes	No
Unreal Engine 3	Yes	Yes	Yes	No	No
Vision Engine	Yes	No	Yes	Yes	No

Table 2

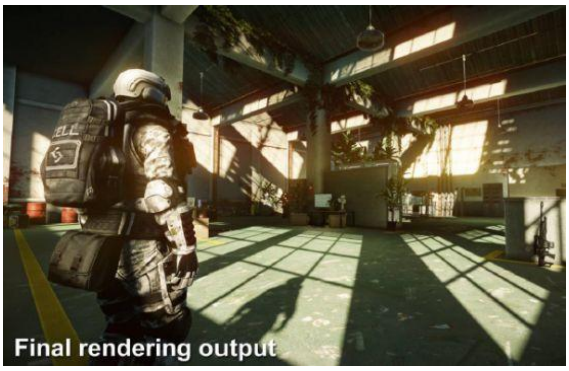

Fig 1.3 Final Rendered output (CE3)

Fig 1.4 Explosion & graphics (Ryse: son of Rome)

Fig: 1.5 CryEngine 3 on console & PC (below)

Fig 1.6 Crysis 3 depth of field using CryEngine 3

1.5.3 Anvil

Company: Ubisoft

As seen in: Assassin's Creed, Prince of Persia, Assassin's Creed II, Assassin's Creed III

Anvil (development project name Scimitar before 2006) is a game engine created in 2007 by video game developer Ubisoft Montreal for use on Microsoft Windows, PlayStation 3, Xbox 360, Wii U, PlayStation Vita, and PlayStation 4. Claude Langlais (Technical Director of Ubisoft Montreal)

says that modelling is done in 3ds Max for environment and ZBrush for characters.[2] The engine uses Autodesk's HumanIK middleware to correctly position the character's hands and feet in climbing and pushing animations at run-time. Anvil has been improved for Assassin's Creed II. Features added include a full night and day cycle, enhanced draw distance, the same vegetation technology used in Far Cry 2, improved lighting, reflection and special effects, new cloth system, and a new AI and NPC navigation system. Assassin's Creed: Lineage short films made by Hybride Technologies (a post-production VFX studio acquired by Ubisoft) and Ubisoft Digital Arts used assets from Anvil to recreate the environment in which the live actors are filmed. The latest iteration of the engine called AnvilNext which is used for the recently released title Assassin's Creed III has support for weather cycling. AnvilNext is reported to have the ability to render massive crowds up to 2000 characters (while it was a few 100s in the previous versions of the engine). This version of the engine also features a dynamic game world, where the game world will change itself over the course of time. This means enemy settlements may appear/disappear relating to the events in the game. AnvilNext will also feature improved visuals and AI.

- Night and day cycle
- New load distance for more detailed open-world environment from greater distances
- The same internal vegetation technology used in Far Cry 2
- Several rendering improvements to support improved lighting, reflection and more special effects
- New cloth system used extensively for characters
- New Artificial Intelligence architecture allowing much more diverse gameplay and mission.
- Enhanced NPC navigation showcasing parkour-like acrobatics that will spice-up the chases and escape missions

Changes to the Anvil pipeline allowed the team to double the number of bones in the face, with a concentration around the crucial areas like the eyes and mouth. Improvements to the cloth and other physical objects also help with make the cinematic close ups look more real. The new system allows the developers to capture body motions, facial performance, and voiceovers for up to six actors at the same time. Instead of stitching these elements together, Ubisoft can capture the true expression of the actors and the scene. This means they are asking for more physical performances from the actors. They need to move about the scene's environment interacting with each other an inanimate object as naturally as possible – all while decked out in black spandex with cameras strapped around their necks. This new system may not produce the uncanny facial animations of Rockstar's L.A. Noire, but the improvements created by AnvilNext puts Ubisoft on par with other revered studios that excel at cutscenes like Naughty Dog and Sony San Diego.

Fig 1.7 Assassin's Creed 3 lighting

Fig. 1.8 Anvil next snowy weather rendering

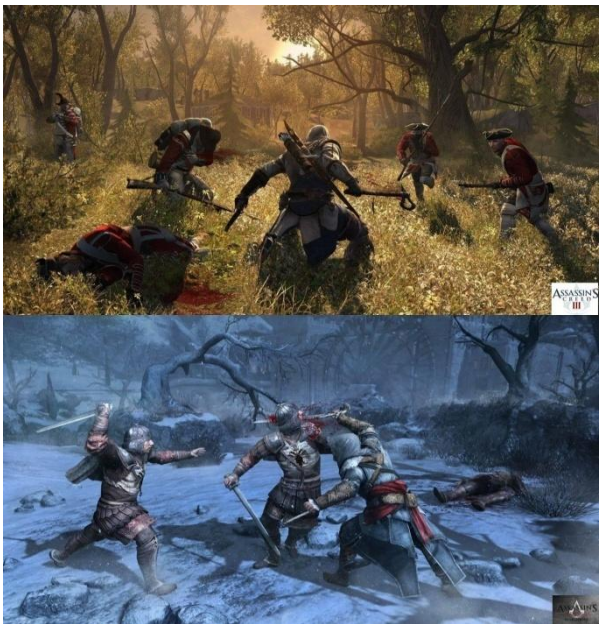

Fig 1.9 Comparison of Anvilnext & Anvil (below)

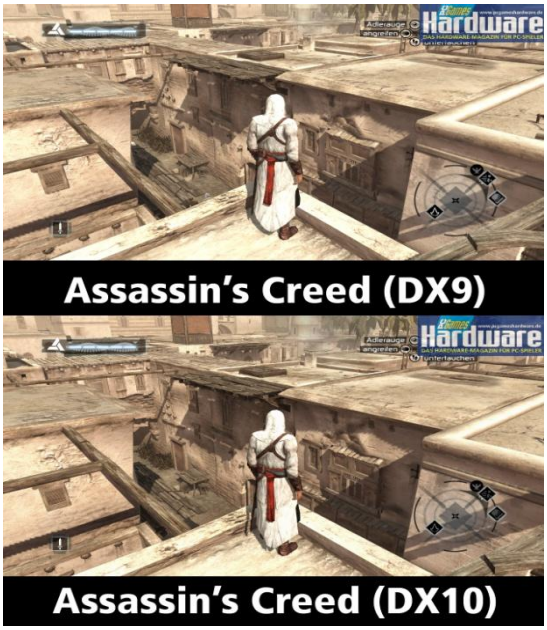

Fig. 1.10 AC direct x 9 compared with direct x 10

1.5.4 Unreal Engine

Company: Epic games

As seen in: Gears of War, Mass Effect, BioShock, Unreal Tournament, Deus Ex, GRAW, Red Steel, Borderlands, Brothers in Arms, Homefront, Mirror's Edge, Singularity, Rainbow Six: Vegas and more.

The Unreal Engine is a game engine developed by Epic Games, first illustrated in the 1998 first-person shooter game Unreal. Although primarily developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, MMORPGs, and other RPGs. With its code written in C++, the Unreal Engine features a high degree of portability and is a tool used by many game developers today. The current release is Unreal Engine 3, designed for Microsoft's DirectX 9 (for Windows and Xbox 360), DirectX 10 (for Windows Vista) and DirectX 11 (for Windows 7 and later); OpenGL for OS X, Linux, PlayStation 3, Wii U, iOS, Android; Stage 3D for Adobe Flash Player 11; and JavaScript/WebGL for HTML5.

1.5.4.1 Versions

- Unreal Engine 1

Making its debut in 1998 with Unreal, the first generation Unreal Engine integrated rendering, collision detection, AI, visibility, networking, scripting, and file system management into one complete engine. Unreal Engine 1 provided an advanced software rasterizer and a hardware-accelerated rendering path using the Glide API, specifically developed for 3dfx GPUs,[4] and was updated for OpenGL and Direct3D. Large parts of the game were implemented in a custom scripting language called Unreal Script. The initial network performance was also very poor when compared to its biggest competitor, Quake II. Epic used this engine for both Unreal and Unreal Tournament. The release of Unreal Tournament marked great strides in both network performance and Direct3D and OpenGL support. The engine became very popular due to the modular engine architecture and the inclusion of a scripting language, which made it easy to mod, including total conversions like Tactical Ops.[6][7]For instance, a developer was able to replace the original renderer from UE1 with a DirectX 10 renderer in 2009.

- Unreal Engine 2

The second version made its debut in 2002 with America's Army. This generation saw the core code and rendering engine completely re-written. In addition, it featured UnrealEd 2, which debuted with the previous generation of the engine and was shortly followed later by UnrealEd 3, along with the Karma physics SDK. This physics engine powered the ragdoll physics in Unreal Tournament 2003 and Unreal Championship. Other engine elements were also updated, with improved assets as well as adding support for the GameCube and the Xbox. Support for the PlayStation 2 console was previously added in UE1. Taking Xbox aside, both GameCube and PS2 were never supported directly by Epic, support being instead farmed out to Secret Level said builds were stale and left behind, the last "official" build PS2 and GC saw was build 927 dated April 2002; last official UE2.5 build was build 3369. As such, third parties looking to use further Unreal Engine revisions had to do their own builds throughout the generation, as they had to in more recent years with the Wii, X360, PS3, PSP, and 3DS. UE2.5, an update to the original version of UE2, improved rendering performance and added vehicles physics, a particle system editor for UnrealEd, and 64-bit support in Unreal Tournament 2004. A specialized version of UE2.5 called UE2X

was used for Unreal Championship 2: The Liandri Conflict on the original Xbox platform. It featured optimizations specific to that console. EAX 3.0 is also supported for sound. Unreal Engine 2.X was build 2227, dated March 2004. On March 24, 2011, Ubisoft Montreal revealed that UE2.5 was successfully running on the Nintendo 3DS.

- Unreal Engine 3

The third and current generation of the Unreal Engine (UE3) is designed for DirectX (versions 9-11 for Windows and Xbox 360), as well as systems using OpenGL, including the PlayStation 3, OS X, iOS, Android, Stage 3D for Adobe Flash Player 11, JavaScript/WebGL for HTML5,[10] PlayStation Vita, and Wii U. Its renderer supports many advanced techniques including HDRR, per-pixel lighting, and dynamic shadows. It also builds on the tools available in previous versions. In October 2011, the engine was ported to support Adobe Flash Player 11 through the Stage 3D hardware-accelerated APIs. Epic has used this version of the engine for their in-house games. Aggressive licensing of this iteration has garnered a great deal of support from many prominent licensees. Epic has announced that Unreal Engine 3 runs on both Windows 8 and Windows RT.

In addition to the game industry, UE3 has also seen adoption by many non-gaming projects, for instance:
The popular children's TV show LazyTown used UE3 during filming to generate virtual sets for real-time integration with footage of actors and puppets performing in front ofgreen screens.
The animation software "Muvizu Play", which was released in April 2013, uses the UE3.
In March 2012, the FBI licensed Epic's Unreal Development Kit to use in a simulator for training.

- Unreal Engine 4

On August 18, 2005, Mark Rein, the vice-president of Epic Games, revealed that Unreal Engine 4 had been in development since 2003. Until mid-2008, development was exclusively done by Tim Sweeney, founder and technical director of Epic Games. The engine targets the eighth generation of PC hardware and consoles.

In February 2012, Mark Rein said "people are going to be shocked later this year when they see Unreal Engine 4". Unreal Engine 4 was unveiled to limited attendees at the 2012 Game Developers Conference, and video of the engine being demonstrated by developer Alan "Talisman" Willard was released to the public on June 7, 2012 via GameTrailers TV. This demo was created on a PC with triple GeForce GTX 580 (tri SLI) and can be run on a PC with a GeForce GTX 680.

One of the major features of UE4 is real-time global illumination using voxel cone tracing, eliminating pre-computed lighting. UE4 also features new developer features to reduce iteration time and allow direct updating of C++ code. New features of the debugger for "Kismet" (a visual scripting engine that debuted in UE3) allow developers to directly visualize code while testing. The developer can then jump to the source code and edit it in Visual Studio. Elements in the game can be clicked on directly to more easily change the game world. This also ultimately results in less of a divide between technical artist, a designer, and a programmer. The result is a reduced time to compile code and allows game creators to tweak settings in real time.
In older engines if you wanted to change the relationship between your weapon damage and how long it'll take to kill a creature, you may spend a couple of days iterating, but if you have to spend a lot of time waiting for a build every time, you're talking one change, waiting 15 minutes for the compile to complete, and then play the game, get to the point where you can test it, test it, exit the game, change, compile. Now, since all of that can be done very quickly within the tools, it's-make the change, play, when it compiles, finish, shoot the guy, and then escape, make the change, play. The iteration time is down to 30 seconds instead of 15 minutes. Our ability to kind of roll through and see how the game is playing out is much faster.

Throughout the lifetime of the UE3, significant updates have been incorporated:

- Epic Games announced at Game Developers Conference (GDC) 2009 some improvements made to Unreal Engine 3. These included:
  - Unreal Lightmass: a global illumination solver. Provides high-quality static lighting with next-generation effects, such as soft shadows with accurate penumbrae, diffuse, specular inter-reflection, and colour bleeding.
  - The ability to add fracture effects to static meshes to simulate destructible environments
  - Soft body dynamics (physics)
  - Large crowd simulation
- In December 2009, Epic demoed UE3 running on Apple's 3rd generation iPod Touch. They said that this will also support iPhone 3GS, and also an unknown mobile platform which has been revealed to be webOS at CES 2010. It has been revealed so far to be something on Nvidia's Tegra platform, and also Palm's webOS running PowerVR's SGX chip.
- In March 2010, Steamworks was integrated into the software, and is offered to licensees.
- In June 2010, Epic Games revealed Epic Citadel, a tech demo to showcase Unreal

Engine 3 on iOS devices (iPhone, iPod touch, and iPad devices).

- In June 2010 during the Electronic Entertainment Expo 2010, Mark Rein (vice president of Epic Games) showcased a tech demo of Gears of War 2 in stereoscopic 3D running on an Xbox 360 thanks to the TriOviz for Games Technology. "This technology's great because it works on normal HD TVs, as well as the very high end 3DTVs," Rein commented toComputer and Video Games. "We're not planning to re-release this in 3D – unless Microsoft want us to – but I'm sure it's technology may be keen to put in the games developed by our partners."
- In October 2010, TriOviz for Games Technology has been officially integrated in Unreal Engine 3, allowing to easily convert in stereoscopic 3D, numerous past and upcoming games developed on Xbox 360 and PS3 with this engine.
- As of March 2011, the Unreal 3 Engine supports DirectX 11. Epic Games showcased it with a real-time demonstration video, entitled "Samaritan". Additions include tessellation and displacement mapping, advanced hair rendering with MSAA, deferred shading with MSAA, screen space subsurface scattering, image-based lighting, billboard reflections, glossy reflections, reflection shadows, point light reflections, and bokeh depth of field.
- In July 2011, Geomerics announced that their real-time global illumination solution Enlighten is now integrated with Unreal Engine 3 and available to licensees.
- In October 2011, Epic Games announced that a version of the engine would be compatible with Adobe Flash Player.
- In May 2012, UE3 added support for the RealD 3D stereoscopic technology.
- In March 2013, Mozilla and Epic Games have demonstrated UE3 running on the browser using HTML5 and JavaScript technologies[7]

Fig. 1.11 Arkham Origins showcasing UE3

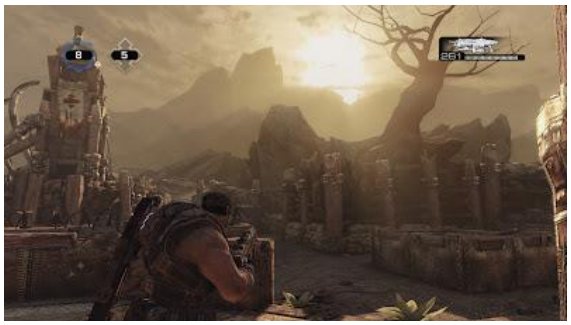

Fig. 1.12 UE3 lighting & environment detail (GoW3)

Unreal engine is by far the most widely used engines in a number of games. This is mainly because it offers the developer a wide range of options to tweak their product. Unreal engine has the ability to be adapted according to the developer's choice thus making it a favourite among them. Unreal engine is available for most of the platforms out there – playstation 3. xbox 360, PC, playstation 4, xbox one, playstation vita, playstation portable etc. Generally due to its tweak able nature this engine renders the best quality on pc platform (fig. 1.14)

In following figure (fig. 1.13) Unreal engine is compared with MSAA (multi sample anti aliasing) on in the above part and with MSAA off in the below part. One can notice the change in the graphical quality.

In the next figure (fig. 1.14) upcoming unreal engine 4 is compared on console (ps4) and pc version. Notice that the pc version looks better.

Fig. 1.13 MSAA on (above) off(below)

Fig. 1.14 Upcoming UE4 on PS4 (left) and PC (right)

1.5.5 Frostbite

Company: DICE

As seen in: Need for Speed: the run, Battlefield 3, Battlefield: Bad Company 2 etc.

Frostbite is a game engine developed by DICE, creators of the Battlefield series. The engine currently is designed for use on Microsoft Windows, PlayStation 3, PlayStation 4, Xbox 360, and Xbox One platforms and is adapted for a range of video game genres. The engine was first used by DICE to create first-person shooters, but it has been expanded to include various other genres such as racing and real-time strategy and is employed by a number of EA studios. Thus far, the engine is exclusive to Electronic Arts with all titles being published by EA. DICE used the first generation of the engine for their in-house games Battlefield: Bad Company, Battlefield 1943, and Battlefield: Bad Company 2. The Frostbite engine launched with the release of Battlefield: Bad Company; subsequent titles employed an updated version referred to as Frostbite 1.5. Frostbite 1.5 was also used for the multiplayer component of Medal of Honor which DICE developed. (The single-player was developed by another EA studio with the Unreal Engine 3.) The next generation of the engine, Frostbite 2, debuted with the release of Battlefield 3. The creation of Frostbite 2 started the first usage of the Frostbite engine by other EA studios other than DICE. Some notable Frostbite 2 powered games include Need for Speed: The Run, Medal of Honor: Warfighter, and Army of Two: The Devil's Cartel. A new version, Frostbite 3, will debut with Battlefield 4.

1.5.5.1 Versions

- Frostbite 1

Frostbite debuted in 2008 with Battlefield: Bad Company. It features HDR Audio, which adjusts different types of sounds' loudness and lets players hear important sounds clearly even if there are other noises being generated (for example, gunshot sounds are always louder than in-game music; the in-game music will lower in volume while shots are being fired), and Destruction 1.0, which allows the player to destroy certain objects, like walls.

- Frostbite 1.5

The second version of Frostbite debuted with Battlefield 1943 in 2009. It improved on the in-game destruction capabilities with Destruction 2.0, allowing the player to destroy entire buildings instead of just walls. In 2010, DICE released Battlefield: Bad Company 2 using this version of the engine, which was the first game to bring Frostbite Engine to the Windows platform. The Windows version of the game has minor DirectX 11 shader support; consequently, it does not take full advantage of the entire API. The multiplayer component of Medal of Honor also uses this version of the engine, though with limited in-game destruction capabilities.

- Frostbite 2

The first major upgrade to Frostbite debuted in 2011 with Battlefield 3. It takes full advantage of the DirectX 11 API and 64-bit processors, with no support for DirectX 9 (nor, therefore, Windows XP). It also features enhanced in-game destruction with Destruction 3.0, creating more refined physics than its predecessor and quasi-realtime radiosity using Geomerics' Enlighten technology. Additionally an important distinction on its naming is that Frostbite 2 is not called Frostbite 2.0.

DICE has given several presentations on advances in their rendering technology with primary focus on the aspects of Animation, Destruction, Scale, Rendering, and Audio:

SIGGRAPH 2010:

- Tile-based deferred shading acceleration" via DirectCompute. This is being ported to PlayStatio3's SPUs.
- Morphological Anti-Aliasing (MLAA), again implemented with DirectCompute, for bandwidth conservation.
- Quasi-realtime radiosity using Enlighten from Geomerics.
- Improved environmental destruction.

GDC 2011:

- Realtime approximated subsurface scattering.

SIGGRAPH 2011:

- Separable bokeh depth of field
- Tile-based deferred shading on Xbox 360
- Temporally stable screen space ambient occlusion (SSAO)

GeForce LAN 6:

DICE's lead graphics guru, Johan Andersson, presents an in-depth examination of Battlefield 3's game engine and visuals.

Other:

Streaming data from disc to memory: We can have 512 megs every hundred metres if we wanted to[8].

- Frostbite 3

Frostbite 3 is the next generation version of Frostbite and is powering the upcoming games Battlefield 4 (set to release October 29th of 2013), Need for Speed: Rivals (to be released in November, 2013), Star Wars: Battlefront (set to release the Q2/Q3 of 2015) and the prequel to Mirror's Edge. The game engine has several upgrades including improved tessellation technology. It also features Destruction 4.0, which enhances the in-game destruction over its predecessors. Frostbite 3 will also power Dragon Age: Inquisition, Plants vs. Zombies: Garden Warfare and the next Mass Effect.

Fig. 1.15 Frostbite 2 in action (BF 3)

Fig. 1.16 First racing game to implement frostbite 2 engine. Improvements on the engine allow better streaming technology which allows larger regions for the game while running at a minimum of 60 frames per second. Further improvements to the audio and lighting engines have been made in this version.

1.5.6 IW (Infinity Ward) engine

Company: Infinity Ward

As seen in: Call of Duty: MW, CoD: Black ops, Black ops II, MW 3

The IW engine is a game engine developed by Infinity Ward for the Call of Duty series and has recently become the main engine for the James Bond video games. The engine itself is proprietary with inclusion of by id Software. It has been used by Infinity Ward, Treyarch, Raven Software, and Sledgehammer Games. The engine has arguably been distinct from the id Tech 3 engine on which it is based since Call of Duty 2 in 2005. The engine's name was not publicized until IGN was told at the E3 2009 by the studio that Call of Duty: Modern Warfare 2 would run on the "IW 4.0 engine". Development of the engine and the COD games has balanced maintenance of 60 frames per second with implementing advanced graphical features. Call of Duty 4: Modern Warfare was released using version 3.0 of the engine. This game included features such as bullet penetration, improved AI, lighting engine upgrades, particle system enhancements and many more improvements. Treyarch began using an enhanced version of the IW 3.0 engine for Call of Duty: World at War. Improvements were made to the physics model and dismemberment was added. Environments also featured more destructibility and could be set alight using the flamethrower. The flamethrower featured propagating fire and it was able to burn skin and clothes realistically. Treyarch, however, then modified the engine for their James Bond title, Quantum of Solace.

Call of Duty: Modern Warfare 2 was released using the IW 4.0 engine, the only game to do so. The IW 4.0 engine featured texture streaming technology to create much higher environmental detail without sacrificing performance. Call of Duty: Black Ops was not based on IW 4.0; rather, Treyarch further enhanced the version of IW 3.0 they had used in their previous game. This version of the engine also featured streaming technology, lighting enhancements, and support for 3-D imaging. Call of Duty: Modern Warfare 3 utilizes an improved version of the IW 4.0

Call of Duty: Black Ops II was developed using a further iteration of the IW engine. Texture blending has been improved due to a new technology called "reveal mapping" which compares tones between two textures and then blends them together. Also, there have been upgrades to the lighting engine which include HDR lighting, bounce lighting, self-shadowing, intersecting shadows and various other improvements. Call of Duty: Black Ops II takes advantage of DirectX 11 video cards on the PC version of the game. The "zombie" mode has been moved to the multiplayer portion of the engine which will allow for much more variety within this part of the game. Black Ops II was released on the 13th November 2012.

Call of Duty: Ghosts has been upgraded from the MW3 engine seen in Call of Duty: Modern Warfare 3. It is unknown at this time whether or not any engine features have been taken from Call of Duty: Black Ops II. The engine used in Ghosts is an engine compatible with next-gen systems such as Xbox One, PS4 and the Wii U so polygon counts, texture detail and overall graphical fidelity has been increased. This game engine will also be compatible with Windows PC, PS3 and Xbox 360. The Ghosts Engine features technology from Pixar, SubD, which increases the level of detail of models as you get closer to them. Mark Rubin has also said about the HDR lighting "We used to paint it in and cover up the cracks, but now it's all real-time". Ghosts will use Iris Adjust tech which allows the player to experience from a person's point of view how their eyes would react to changes in lighting conditions realistically. Other features include new animation systems, fluid dynamics, interactive smoke, displacement mapping and dynamic multiplayer maps. Ghosts is scheduled to released on November 5, 2013.

1.5.6.1 Games showing IW features (Table 3)

	Engine Name	Release Year	Upgraded from	Change summary
Call of Duty 2	IW 2.0	2005	Modified version of id Tech 3 from Quake III Arena	• Normal mapping • Light bloom • Heat haze • Unified shadowing system
Call of Duty 4: Modern Warfare	IW 3.0	2007	Call of Duty 2's IW 2.0 engine	• Bullet penetration • Improved AI • Lighting engine upgrades • Particle system enhancements • "Ragdoll" physics integrated into character death animations • Self-shadow every frame
Call of Duty: World at War	IW 3.0	2008	Call of Duty 4: Modern Warfare's IW 3.0 engine	• Improved physics • Dismemberment • More destructible environments • Realistic burning of skin and clothing • Propagating fire
Quantum of Solace	IW 3.0	2008	Call of Duty 4: Modern Warfare's IW 3.0 engine	• Unknown
Call of Duty: Modern Warfare 2	IW 4.0	2009	Call of Duty 4: Modern Warfare's IW 3.0 engine	• Texture streaming technology
Call of Duty: Black Ops	IW 3.0	2010	Call of Duty: World at War's IW 3.0 engine	• Texture streaming technology • Lighting enhancements • 3-D stereoscopic support.
Call of Duty: Modern Warfare 3	MW3 engine	2011	Call of Duty: Modern Warfare 2's IW 4.0 engine	• Improvements to texture streaming technology to allow for larger regions. • Lighting engine enhancements to show reflections of some objects on tile floors. • Improvements to the audio engine
Call of Duty: Black Ops II	Black Ops II engine	2012	Call of Duty: Black Ops's IW 3.0 engine	• Reveal mapping (improved texture blending) • Water effects • Improved lighting effects • Lens flare effects • HDR lighting • Bounce lighting • Self-shadowing • Intersecting shadows • DirectX 11 for PC version.
Call of Duty: Ghosts	Next Generation Call of Duty Engine/Ghosts Engine	2013	Call of Duty: Modern Warfare 3's MW3 engine	• Pixar's SubD (models sub-divide the closer the player gets to them)[6] • New animation systems for movement (sliding, leaning etc.)

				• Fluid Dynamics • Interactive Smoke • Improved AI • Dynamic multiplayer maps • Displacement mapping • True real-time HDR lighting • Iris Adjust tech • Graphics Tessellation For PC, PlayStation 4 and Xbox One

Fig. 1.17 Modified IW 3.0 engine in Black Ops 2

Fig. 1.18 Upcoming CoD ghosts next gen IW engine

VII. COMPARISON

Comparison between various game engine is not a simple task because of their various genre, types, multimedia support, middleware support, language and platform dependencies, rendering techniques and many other sub features[9].

- IW engine

The IW Engine is a game engine developed by Infinity Ward, based on id Tech 3, which was first used on the highly successful game Call of Duty 2. Since then, it has been continually upgraded and used on the games Call of Duty 4, Call of Duty: World at War, Call of Duty: Modern Warfare 2, and Call of Duty: Black Ops.

The engine features the following technologies:

* True world dynamic lighting
* HDR lighting effects
* Dynamic shadow
* Depth of field
* Dynamic physics system
* Radgoll physics
* Realistic fire
* Advanced weather effects
* Realistic character animation
* High detailed objects

Table 4: Comparable features of Infinity ward game engine

- Unreal Engine 3

Main Features

- 64-bit color High Dynamic Range rendering pipeline. The gamma-correct, linear color space renderer provides for immaculate color precision while supporting a wide range of post processing effects such as light blooms, lenticular halos, and depth-of-field.
- Support for all modern per-pixel lighting and rendering techniques including normal mapped, parameterized Phong lighting; custom artist controlled per material lighting models including anisotropic effects; virtual displacement mapping; light attenuation functions; pre-computed shadow masks; directional light maps; and pre-computed bump-granularity self-shadowing using spherical harmonic maps.
- Advanced Dynamic Shadowing. Unreal Engine 3 provides full support for four shadowing techniques:
  - Dynamic stencil buffered shadow volumes supporting fully dynamic, moving light sources casting accurate shadows on all objects in the scene.
  - Dynamic characters casting dynamic soft shadows on the scene using shadow buffers. Shadow buffer filtering takes samples on a jittered disc that are rotated per-pixel to detect shadow penumbras. Dynamic branching is then used to refine shadow coverage in penumbra regions.
  - Ultra high quality and high performance pre-computed shadow masks allow offline processing of static light interactions, while retaining fully dynamic specular lighting and reflections.
  - Directional Light Mapping enables the static shadowing and diffuse normal-mapped lighting of an unlimited number of lights to be precomputed and stored into a single set of texture maps, enabling very large light counts in high-performance scenes.
- All of the supported shadow techniques are visually compatible and may be mixed freely at the artist's discretion, and may be combined with colored attenuation functions enabling properly shadowed directional, spotlight, and projector lighting effects.
- Volumetric environmental effects including height fog.
- Full support for seamlessly interconnected indoor and outdoor environments with dynamic per-pixel lighting and shadowing supported everywhere.
- *UnrealKismet*, our visual scripting system:
  - Gives artists and level designers virtually limitless control over how a level will play without touching a single line of code.
  - By connecting together simple events and actions created by programmers, everything from simple behaviours to complete gameplay prototypes can be assembled quickly.
  - *UnrealKismet* supports hierarchies of scripts for organizing very complex sequences into manageable units.
- The Unreal Editor (UnrealEd) is a pure "What You See Is What You Get" content creation tool filling the void between XSI, 3D Studio Max and Maya, and shippable game content.
- A powerful browser framework for finding, viewing, and organizing game assets of all types.
- Visual placement and editing of gameplay objects such as players, NPCs, inventory items, AI path nodes, and light sources - with a full realtime view of their appearance, including 100% dynamic shadowing.
- Includes a data-driven property editing framework, allowing level designers to easily customize any game object, and programmers to expose new customizable properties to designers via script.
- Realtime terrain editing tools allowing artists to elevate terrain, paint alpha layers onto terrain to control layer blending and decoration layers, collision data, and displacement maps.
- Visual *Material Editor*. By visually connecting the color, alpha and coordinate outputs of textures and programmer-defined material components, artists can create materials ranging from simple layered blends to extremely complex materials and dynamically interacting with scene lights.
- Animation tool enables artists to import models, skeletons, and animations, and to tie them to in-game events such as sounds and script notifications.
- In-editor "Play Here" button puts gameplay just one mouse click and a fraction of a second away. Here, you can test gameplay in-editor in one window while modifying objects and rearranging geometry in another.
- Plug-ins for 3D Studio Max and Maya to bring models into the Unreal engine with mesh topology, mapping coordinates, smoothing groups, material names, skeleton structure, and skeletal animation data.
- Fully integrated source control, so that artists and level designers can check out content packages, modify, and check in from within the editor.
- All the other niceties you'd expect from a modern content editing tool: Multi-level undo/redo, drag-and-drop, copy-and-paste; customizable key and color configuration; viewport management.

Table 5: Comparable feature of Unreal Engine 3 in a tabular form

- CryEngine 3[10]

Tessellation

CryENGINE®3 now supports DX11 hardware accelerated tessellation on all meshes, including animated characters. Artists can now enable tessellation in real-time! CryENGINE®3 features support for three different types of tessellation, all of which can be mixed so as to significantly increase the quality of asset rendering in-game. With displacement mapping CryENGINE®3 can now represent silhouettes and complex geometric shapes, previously impossible with current generation bump mapping, through displacement mapping.

The other two tessellation methods are designed for mesh refinement, taking a coarse, or low-poly model and using tessellation to create an higher-fidelity and much smoother mesh. As these methods are all editable in real-time in CryENGINE®3 SandboxTM, artists can easily decide which assets should use such features, finding an ideal compromise between visual fidelity and performance.

Real-time Local Reflections (RLR)

Reflections are one of the biggest challenges in real-time rendering, particularly for deferred rendering and lighting engines. Before DX11, only reflections for surfaces such as water; or environment probe reflections have been possible. CryENGINE®3 now introduces a new method: Real-time Local Reflections (RLR). RLR approximates ray-traced HDR reflections local to objects. This technique allows for nd of curved surface in the scene to reflect the nearby surroundings efficiently in real time. This includes immensely complex objects with self-reflections, which are not feasible with cube-map or planar reflections.

Parallax Occlusion Mapping (POM)

Parallax Occlusion Mapping (POM) returns in CryENGINE®3, optimized for all platforms and can now be used to add self-shadowing and the appearance of depth on all surfaces. CryENGINE®3's POM efficiently approximates surface displacement in tangent space. Due to it's low cost on all platforms, self-shadowing support and simulation of macro details, POM grants a viable alternative to tessellation on all surfaces.

Advanced Tone Mapping with HDR S-Curve

The CryENGINE® tone-mapper is improved and updated to give users S-Shaped curve color control, which better mimics the exact behavior of film tone reproduction. This allows for a wider tonal range, improved definition on the dark tones, giving a richer color space for CryENGINE®3 games.

Ocean and Water Rendering

CryENGINE®3's trademark, benchmark setting water rendering just got better! All water volumes can now use DX11 hardware tessellation. Mesh geometry is dynamically tessellated, varying by the distance from the camera. This allows for a very high amount of geometry details close to the camera and eliminates geometry aliasing. Shading has also been updated: now supporting sub-surface scattering and wave crest foam approximation. CryENGINE®3 water also offers a more accurate surface normal computation, even allowing shader based ripples to interact with the CryENGINE® physics system.

Natural lighting & dynamic soft shadows

CryENGINE®3 features near-reality natural lighting at a low cost on multiple platforms, and creates soft shadows that dynamically respond to natural movements in real-time. High-resolution, perspective-correct and volumetric smooth-shadow implementations are also included in CryENGINE®3.

Deferred lighting

The unique CryENGINE®3 deferred lighting solution allows the efficient rendering of a vast number of light sources with per-pixel shading. It can also considerably simplify engine performance tuning across multiple platforms.

SSDO-Screen Space Directional Occlusion and Contact Shadows

Crytek introduced SSAO in real-time gaming and now enhances it with the new technique of Screen Space Directional Occlusion (SSDO). SSDO allows any number of lights to be included in the calculation of Ambient Occlusion (AO); and for the color of all lights to be taken into account. SSDO improves physical lighting accuracy and provides dynamic contact shadows, even from the efficient CryENGINE®3 deferred ambient lights. The new CryENGINE®3 DX11 renderer allows contact shadows to be cast from every single light source in the game environment!

CryENGINE® 3 SANDBOX™ - THE ULTIMATE MULTI-PLATFORM TOOLSET

CryENGINE®3 Sandbox™ gives developers full control over their multi-platform creations in realtime. It features many improved efficiency tools that enable the fastest development of game worlds and game-play available on PlayStation®3, Xbox 360™ and PC. All features of CryENGINE®3 games can be produced, edited and played immediately with the "What You See Is What You Play" (WYSIWYP) system.

Below, you'll find just a fraction of the features available in CryENGINE®3 Sandbox™!

Flow graph

CryENGINE®3's simple yet powerful visual editing system in gives designers intuitive interfaces to create and control events, triggers and game logic. This lets them build game play systems and complex levels without needing to write a single script.

TRACK VIEW EDITOR

The Track View is the embedded Sandbox cut scene editing tool for making interactive movie sequences with time-dependent control over objects and events in CryENGINE®3. Creating cinematic cut scenes and scripted events are both possible, allowing you to sequence objects, animations, and sounds in a scene that can be triggered in the game, and played either as a detached cutscene from the third person view, or from the first person perspective of the player in-game.

Fully flexible time of day system

The time of day can dynamically change during a game mission to reflect the position of the sun and moon, as well as lighting and atmospheric conditions over any predefined day/night cycle, from a foggy morning sunrise to a fiery sunset or a clear moonlit night. Create visually rich time-of-day simulation for your game world or its locations and achieve the most realistic or surreal settings.

Advanced terrain system with integrated voxel objects technology

CryENGINE®3 features the only real-time terrain tool-set that enables the creation of stunning concave environmental features. This lets designers place over-hanging cliffs, caves or tunnels seamlessly in each level.

Multi-Layer Navigation Mesh

The Multi-Layer Navigation Mesh (MNM), is an enhanced dynamic navigation data structure used by AI Agents to path-find through levels. This system is integrated in CryENGINE®3 SandboxTM; designers can simply use the tools to define areas where the navigation mesh should be generated and let the CryENGINE®3 MNM do the rest of the work! MNM also supports real-time dynamic updates in game, so that when an area changes, the navigation mesh updates without delay.

Multi-core support

To get the most out of modern multi-core processor architectures, CPU-intensive sub-systems of CryENGINE®3 such as graphics, physics, AI, networking and sound, have been optimized to support multiple processors, including innovative support for PlayStation®3 SPU.

Uber-shader technology

CryENGINE®3 shaders are written once in a high-level language, then compiled automatically to each platform. The shader output is optimized based on the artists' settings and the 3D environment in which the shader is used. This creates unique, high-performance effects, such as "cloaking" or wet, muddy and frozen surfaces, which can be layered together and combined with other shaders such as metal, glass and a full range of similar visual effects. CryENGINE®3 supports real-time per-pixel lighting, bumpy reflections, refractions, volumetric glow effects and animated textures to simulate effects including windows, bullet holes and shiny surfaces. It also takes advantage of the unified shader architecture of current and next-gen hardware. The combination of deferred lighting and this technology lets CryENGINE®3 avoid the traditional shortcomings of solely using uber shaders.

Interactive & destructible environments

All environments in CryENGINE®3 can be dynamically physicalized, regardless of their nature (wood, steel, concrete, natural vegetation, cloth and soft body physics). This allows procedural destruction and deformation of as much of the environment as the game requires. All broken objects and parts can be interactive, with realistic properties such as mass or buoyancy applied to all kinds of debris.

Table 6: Comparable feature of CryEngine 3

- Frostbite 2

Frostbite 2 was one of the first engines released to support DirectX 11 technology debuting with DICE's blockbuster Battlefield 3. It takes full advantage of the DirectX 11 API and 64-bit processors, with no support for DirectX 9 (nor, therefore, Windows XP). It also features enhanced in-game destruction with Destruction 3.0, creating more refined physics than its predecessor and quasi-realtime radiosity using Geomerics' Enlighten technology. Additionally an important distinction on its naming is that *Frostbite 2* is not called *Frostbite 2.0*. *Frosbite 2.0 feature list:* • *Improved high-dynamic range (HDR) audio from that seen in earlier incarnations of the Frostbite engine. Sound effects were recorded from different distances to make the audio as realistic as possible.* • *Intended for use on 64-bit CPUs and OSes (Windows Vista or Windows 7 64-bit editions).* • *No support for Windows XP or DirectX 9; DX10 will be supported but the engine is intended for DX11 graphics cards.* • *DirectCompute tile-based deferred shader rendering. Improves performance and graphics.* • *Destruction 3.0, improving on the destructible environments seen in BC and BC2 (Destruction 1.0 and 2.0 respectively).* • *DirectCompute Morphological Anti-Aliasing (MLAA). DX11 only; improves performance and graphics.* • *Real-time radiosity lighting engine, based on the Geomarics Enlighten rendering software. Allows for fully dynamic, fully HDR (high-dynamic range) realistic lighting.*

Table 7: Comparable features of Frostbite 2 engine

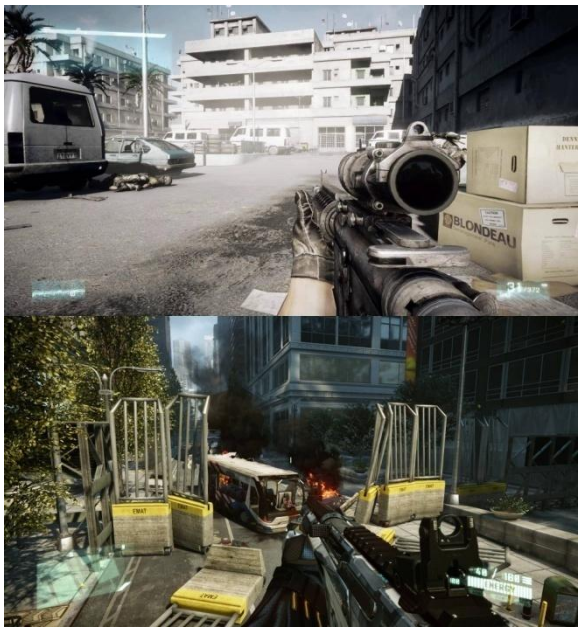

Fig. 20 Comparison Frostbite 2(A) Cryengine 2 (B)

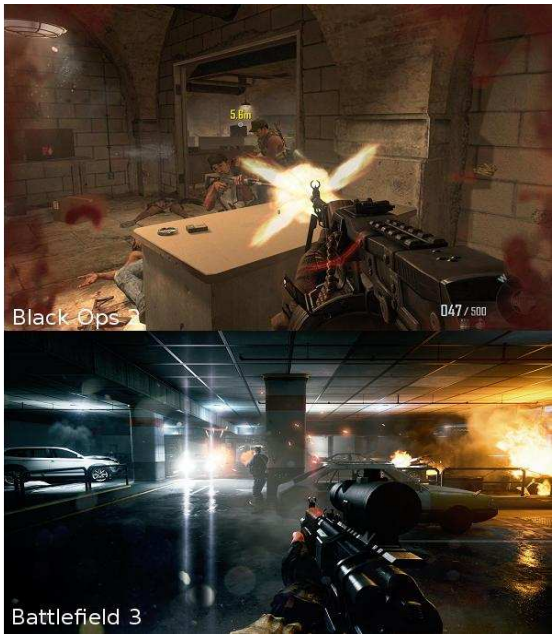

Fig. 21 Comparison IW engine(A) frostbite 2 (B)

Fig. 22 Comparison Cryengine 3 (L) Unreal engine 3(R)

Fig. 23 Comparison Anvil engine (L) RAGE (R)

I. 1.7 CONCLUSION

Clearly there's a lot more to a game than just player mechanics, cameras, and AI. Some games have drivable vehicles, implement specialized types of weaponry, allow the player to destroy the environment with the help of a dynamic physics simulation, let the player create his or her own characters, build custom levels, require the player to solve puzzles, or… Of course, the list of genre- and game-specific features, and all of the specialized soft ware systems that implement them, could go on forever. Gameplay systems are as rich and varied as games are. In this paper I have discussed in somewhat detail possible the main contenders in the market of game engines today which are use in favourite game titles all over. This paper was compiled keeping in mind the need for a comparative study of some of the game engines driving the main games of the market. The engines

considered are only of some of the recent games. Many engines have been lest out. The reason for this is that they couldn't be reviewed here. Video game industry is a very huge one, so are the game engines. The centre of discussion is mainly the graphics rendered by these game engines resulting in the rendered quality of finished product.

REFERENCES

[1] Jason Gregory, *Game Engine Architecture* (A K Peters/CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742)
[2] Tomas Akenine-Moller, Eric Haines, and Naty Hoff man *Real-Time Rendering (3rd Edition).* Wellesley, MA: A K Peters, 2008
[3] James Arvo (editor) *Graphics Gems II.* San Diego, CA: Academic Press, 1991
[4] David H. Eberly *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics.* San Francisco, CA: Morgan Kaufmann, 2001
[5] Lilly, Pauly *Doom to Dunia: AVisual History of Game Engines* Maximum PC (July 17, 2009)
[6] http://en.wikipedia.org/wiki/Game_engine
[7] Ian Millington *Game Physics Engine Development* San Francisco, CA: Morgan Kaufmann, 2007
[8] Aaron Lee *The top 14 game engines* 22nd March 2013
[9] IB Times staff reporter *International Business Times* (June 22, 2012)
[10] *CryEngine 3 technology* retrieved from (http://mycryengine.com/index.php?conid=5)