The Evolution of First Person Shooters

An Honors Thesis (HONRS 499)

By

Joseph McKay

Thesis Advisor
Adrienne Jones

Ball State University
Muncie, Indiana

December 2003

December 21, 2003
Abstract:

Video games have become an integral part of popular culture over the past decade. Gaming is surpassing TV and film as a favorite pastime. Last year the gaming industry made $9.4 billion dollars in revenue. Despite the enormous draw of video games, they aren’t given the prestige and academic attention that film receives. This thesis covers the evolution of the First Person Shooter from game genre conception to its present state in order to demonstrate the validity of studying the video game medium. Since its beginning in 1992, the First Person Shooter has had a tremendous impact on game development. The genre has consistently been at the forefront of cutting edge technology and continues to revolutionize the way games are played and viewed.
Acknowledgments:

I would like to thank my advisor, Adrienne Jones for her constant support and advising throughout the duration of this thesis. Without her help I would have never been able to put my passion into writing.

I would also like to thank my friends Evan, Amanda, Stephanie, David and, Jeff for sticking by, dealing with my gaming rants, and imparting their knowledge.
Table of Contents

Introduction ................... The First Person Shooter

Chapter 1 ....................... Birth of a Genre

Chapter 2 ...................... The Father of the FPS has Arrived

Chapter 3 ....................... The Clone Wars Part I

Chapter 4 ....................... Established Dominance

Chapter 5 ....................... Unreal Competition

Chapter 6 ....................... The Future is Now

Chapter 7 ....................... Multiplayer Mayhem

Chapter 8 ....................... Genre Breakers

Chapter 9 ....................... Rise of the Console FPS

Chapter 10 ..................... The Clone Wars Part II

Chapter 11 ..................... What does the Future Hold

Bibliography
Introduction
The First Person Shooter

In the gaming industry no genre is as popular as the First Person Shooter (FPS). Spawned by a group of independent programmers and game designers in 1992, it has remained one of the most copied and reproduced genres for the past ten years. What makes a game an FPS, why it is so popular, and how it has grown from one game to several hundred. This thesis presents a brief history of where the FPS began, its major players, and how it evolved as a genre and impacted the rest of the gaming scene and industry.

The main characteristic of the FPS is that it takes place from a first person point of view. That is, the player is experiencing the game from the eyes of the main character who they are controlling. Everything that happens in the game unfolds in front of the player’s eyes and the action is centralized around them. The enemies aren’t just attacking or talking to the character, they are interacting with the player. This gives those playing a much closer involvement than a 2D side scroller or a 3rd person game (camera follows the character onscreen) where the events happen to a character, not a player. This kind of viewpoint allows for a level of immersion unique to the FPS.

Nearly every FPS also has a standard Heads Up Display (HUD). The character’s weapon or item is present on the bottom of the screen in either the left or righthand side. The other information displayed can be different for each FPS. A player’s health, armor level, and ammunition remaining is usually positioned near the weapon. Some FPS games place this information at the top right of the screen leaving the bottom open for an
inventory screen. Many FPSs also place radar or a miniature map on the top left side to help the player orient themselves and locate enemy positions in the game. In special cases (which depend on the game’s setting) developers place unique pieces of information on the screen that a player needs for that particular game. One such example is *Thief*, where on the bottom middle of the screen is a light meter telling the player how visible they are to opponents.

The core play of any FPS is intense and violent action, with only a few notable examples that will be covered. In many cases, story and character development are placed on the back burner in order to concentrate on gun play. For years the only other activity players had to be concerned about was finding keys to the exit of a level. Early on, FPSs gained the reputation of being mindless action fests (which was and remains generally accurate) but advancements in technology and licensing allowed game makers to concentrate on simply the traditional ‘run and gun’ style play. One thing that has remained true to this day is that FPS games are usually on the forefront of graphics, sound, memory, and other computer technology. The latest and most advanced game engines are made into First Person Shooters and push players to upgrade their PC and console gaming equipment.

Of all the game genres available in the gaming industry, the FPS continues to top charts and excite players and industry critics. However the excitement is not always realized and this genre generated by advance advertising is plagued with clones and mediocre attempts to capitalize on the most successful games. No other genre has met with such a flood of games that promise much, but deliver little. The evolution of the FPS genre has been and continues to be rocky, but is always exciting. The primal soup
that spawned the First Person Shooter was Madison Wisconsin and a group of talented, ambitious, independent developers known as id Software.

Chapter 1

Birth of a Genre

What many gamers of today don’t realize is that the perspective of their favorite FPS games did not originate with id Software. The game Battlezone is widely attributed to be the first FPS and ancestor to Doom (Demaria, p59). Created by Ed Rotberg for Atari in 1974 for the booming arcades, Battlezone put the player in the role of a tank gunner. The experience was brought to life through the use of vector graphics, which uses straight lines to draw geometric shapes. Players were thrust into a realistic landscape and had to fend off enemy tanks in a scenario that translated into a very tense experience for gamers. In fact, the experience was so impressive that the military contracted Rotberg, despite his opposition, to create a version of the game to train soldiers. The military version contained many new features that the entertainment one didn’t. It was programmed with realistic trajectories, a variety of ordinance, friendly and unfriendly vehicles, more magnification levels, and a more realistic control system (Demaria, p59). Many subsequent games used the same perspective, but aren’t FPSs. These included Atari’s Star Wars arcade game and countless flight and racing simulators.

In the year 1992 the guys at id Software released a game that would spawn the FPS genre and take the industry to a place it had never been before, that game was the uber-classic Wolfenstein 3D. It was unlike anything gamers had ever seen before; fast,
brutal, and pleasantly addictive. *Wolfenstein*’s quick gameplay revolved around its lightning fast 3D graphics engine created by John Carmack, id cofounder and lead programmer. To keep players enthralled with the game, the speed of the game engine was designed to keep them in a constant state of adrenaline rush. To keep the game pace consistent, the designers tore out any features that slowed it down. At one time during development, a feature allowed the gamer to search and drag bodies but it was cut because it slowed the game down too much (Kushner, p97). Although the game has 3D in the title, it was really just a hybrid of three dimensional space and two dimensional characters. The game space was 3D, but the characters and objects were 2D sprites (two dimensional characters drawn out and animated like cartoons). The lead artist had to draw every frame of these sprites in several different positions to achieve the illusion of a total three dimensional world. id was so successful in this creation, *Computer Gaming Monthly*, a respected magazine in the industry, hailed it as “the first game technologically capable of...immersing the player in a threatening environment...” (Kushner p114).

Beyond the nuts and bolts of the game was its true draw, the content. *Wolfenstein 3D* pitted the Allied soldier William J. ‘B.J.’ Blazkowicz against Hitler and his Nazi Army. Up until this point, game designers tended to steer clear of controversial subject matter in order to prevent stirring up trouble with society. id broke that barrier with *Wolfenstein*. But the shock value didn’t stop with the inclusion of the Third Reich.

To fill orders for German customers id was forced to send the game in discrete packages due to Germany’s laws prohibiting any content in entertainment that made reference to Nazis. *Wolfenstein* was the first game to show the violent deaths of characters onscreen. When the player killed a Nazi or dog, the character screamed and
bled all over the floor. In preceding games, when characters were eliminated they simply disappeared from the screen. There was no sense of realism associated with the act of killing. id introduced the custom of adding realistic and violent deaths to increase not only the shock value, but also the attraction of high intensity gaming. John Romero, one of id’s cofounders, sums up *Wolfenstein 3D*’s magnetic draw:

> With *Wolfenstein*, the shock was only half of the attraction. The main draw was the super-fast 3D rendering engine and movement. Most of the raves came from the pure adrenaline rush of speeding at 70 frames per second through corridors mowing down Nazis (Kent, p458).

*Wolfenstein 3D* not only paved the way for the FPS genre, but also widely popularized a method of distributing games call Shareware. With Shareware, portions of the full game were downloaded from the internet or sold in stores for a minimal price. If gamers liked what they had played, they could order the full game and pass along the demo to friends. The idea behind Shareware was to maximize the distribution of the product. Get the game into as many gamers’ hands as possible in order to promote the game by taking advantage of free advertising (mostly) and word of mouth. This also allowed many small unknown software developers with little money for advertising or publishers to get their product in view of the market. With the huge success of *Wolfenstein* many companies felt more comfortable with using Shareware and more gamers used it to try out software.
Chapter 2

The Father of the FPS has Arrived

For id’s next game, they grabbed the bar that *Wolfenstein* had set and tossed it sky high. In December of 1993 the world of gaming was forever changed when id released *Doom*. *Doom* expanded on everything that made *Wolfenstein* a hit. It’s widely considered the best game of all time due to its overwhelming success and achievements that set the benchmark for gaming for years to come. *Wolfenstein*’s environment can best be described as running around in a huge, well lit box. It had tiled walls and a limited ceiling giving it an indoor and confined feeling. The engine that powered *Doom* ripped the box apart and exposed the outside world. *Doom* had huge outdoor environments with textured skies and indoor levels with varying ceiling levels. There was no cramped feeling, unless done intentionally by the level editors for suspense and tension.

Another feature that set *Doom* apart was its new animation method and texture mapping. While *Wolfenstein*’s 2D sprites were drawn, the monsters in *Doom* were done with a mixed media approach. Every character was modeled in clay and shot in several different positions with film. Those images were scanned into the PC, colorized, and then programmed as extremely lifelike digitized characters (Kushner p134). The arm for the main character was a digitized version of the artist’s arm holding the various weapon models the team employed for the game. By comparison, the look of *Wolfenstein* was repetitive and bland. To fix that issue and make the world more realistic, id used a method called texture mapping to color the game’s environments. Texture mapping is a lot like wallpapering a house. With this method images are scanned into the computer
and the level designer places them over walls, floors, objects, skies, anything that the designer wants. Scans of a brick wall could be used to tile buildings within the game. *Doom* was one of the first games to use this method and used it extremely well. The world changed practically overnight from the artificial realm to a perfect mixture of organic and mechanical textures. More dynamic lighting was another effect that id introduced. Every game has some kind of lighting, but it’s usually uniform and static. The lights in *Doom* blinked on and off in strobe effects, illuminated halls, and created pitch black corridors. This helped to create a very mysterious and terrifying atmosphere. A player could run into a room only to have the lights suddenly turn off and hear monsters growling nearby in the dark. It’s definitely one of the top reasons why *Doom* was considered such a revolution; the world actually pulled the gamer into it.

id had clearly shocked the industry with its inclusion of Nazis and realistic violence. *Doom* took that shock value and expanded it. The adversaries in *Doom* were not even human, well not anymore, they were the minions of Hell. Using new animation techniques, id’s artists created some of the most terrifying demons ever thrown into entertainment. They were guaranteed to make your heart jump into your throat and came with death animations that made the Nazi’s deaths look like leaking ketchup packets. With these minions and zombies also came other more horrific imagery. Levels were decorated with satanic symbols, brutally mutilated corpses, bleeding walls, and a host of other grotesque visions. The premise of the game was based on a short textual blurb (the extent of game stories so far) of a lone soldier trying to stop an invasion from Hell. The game world looked as close as one could get to a real demonic overrun.
At this time PC gaming was still trailing behind. The console game systems like *Nintendo* and *Genesis* that were hugely popular. One feature of *Doom* in particular helped shift that balance to the PC side. *Doom* allowed two players to connect their PCs via a modem and go head to head against each other. Similarly, the game also supported four players over a Local Area Network (LAN) to play cooperatively or against one another. Almost overnight dorms and offices had their Internet bandwidth crippled due to hordes of people playing online *Doom*. Several companies had to wipe it from their servers and universities had to plead with students to stop playing *Doom* online. The multiplayer aspect was one of the main factors in making *Doom* so popular. One of its creators, John Romero, is credited with coining the phrase ‘deathmatch’ for describing the head to head play (Kushner 150). For the first time PC owners had a game that pitted them against each other. *Doom* set a very high standard for games to come in the following years.

One of the principles that id started with was the importance of free code. Most of id’s employees started out as gamers and hackers, especially John Carmack, the lead programmer who created all id’s game engines. Interestingly, this idea of free code was the main factor why many of id’s employees refused offers from Bill Gates to join *Microsoft* (id was also making a lot more money at this time then Gates was (Kushner p196)). id openly allowed gamers to create modifications to their games, as long as the mods (shorthand term the gaming industry uses for game modifications) didn’t work with the shareware version and make note that the mod is not an id Software product. Within weeks of *Doom*’s release gamers flooded the Internet with new levels from hacked together level editors to new content complete with characters and sounds. By its end
there were *Doom* Barney, *Doom* Simpsons, *Doom* Clintons, *Doom* Star Wars and *Star Trek*. One popular mod replaced the game’s music with tunes from Nirvana, a highly popular band at the time. This trend of customization continued through *Doom*’s lengthy prominence and greatly contributed to its longevity. Even today many games are released with map and mod tools for interested parties to make their own content. It has even helped a majority of the gamers in the mod community to obtain jobs with game developers.

If one wasn’t a part of the gaming community or computer culture of the time, it can be hard to imagine the sheer popularity of *Doom* and what it did for the gaming industry. People played all through work, skipped classes, held tournaments and mod parties, and high profile celebrities like Trent Reznor (lead singer of Nine Inch Nails, another top band) ignored fans, beer, and women to play it (Kushner p159). Gamers literally bowed before id members at conferences. Id’s phones were jammed all the time with job offers and people with ideas on how to make more money off of *Doom*. Bill Gates even saw *Doom* as one of the most important marketing aspects of the new *Windows 95* operating system (Kushner p197). It wasn’t just a game. It was a culture. *Doom* gave Gamers the most exciting, immersive, and terrifying experience no one had ever seen. It was a game that couldn’t be explained, it had to be experienced. As the saying goes “imitation is the sincerest form of flattery.” *Doom* continues to be flattered today with no end in sight.
After the overwhelming success of id's *Doom*, every game developer and publisher sought to place their own FPS on the market. The guys at id had created the ultimate formula and started a whole new genre in the gaming industry. The temptation to capitalize on that formula was impossible to resist. Store shelves and the Shareware medium became saturated with FPS titles. Out of all the games that were released, a few added their own twists to the genre and turned out surprisingly well. Most however would be doomed to be banished to the forgotten pile. The first Clone War had begun.

Ironically one of the first software developers to try and capitalize on the new *Doom* phenomenon was id software itself. Their answer to *Doom*? Why, *Doom II* of course. What did they add to try and top their earlier product? Not much actually. There is an old adage that is embraced in the game industry as both a positive and a negative; if it isn’t broke, don’t fix it. These early years in the FPS history forever engrained that ideology into game culture. *Doom II* was basically the same as *Doom* with new levels, a few new demons, and the addition of the double-barrel shotgun (probably the most used FPS weapon. If the setting is even remotely modern, the game has a shotgun.). The graphics were largely the same as *Doom*, but the environments were more open: *Doom II* took place in the cities of Earth, rather than a confined base on Mars. The idea was still to shoot everything in the level while trying to search for keys to exit, a gameplay style that would be used for years to come, and lingers even today.

Despite the fact the game was largely the same as its predecessor, *Doom II* was immensely popular. It was an instant hit the day it hit the shelves. Many complained the
game wasn’t as addictive or enthralling as the original but that didn’t stop them from multiplaying. *Doom II*’s release also started the death of shareware. *Doom II* was a straight retail release with no free episodes to try before you buy. Games now were beginning to be sold solely on promotion and hype instead of solid gameplay. Around this time was also the arrival of the Entertainment Software Ratings Board (ESRB) ratings. *Doom II* was the first game to receive a mature (suitable for ages 17 and up) rating (Kushner p171).

*Doom* is credited by industry critics starting the online deathmatch craze but as popular as it was, *Doom II* was more popular with the online multiplaying crowd. In 1994 when *Doom II* was released the Internet was beginning to rise to dominance, so the result is not so surprising. People created more levels, mods, and conversions than they did for *Doom* and passed them out over the net. Gamers made levels of their homes, schools, and offices then passed them out to friends. At one point Wizard Works, a Minnesota publisher released *D!Zone* which was a collection of over nine hundred user made mods. When it came out it beat *Doom II* in sales. (Kushner p194) *D!Zone* wasn’t the only new *Doom* content released, id responded by putting out *The Master Levels*, as well as the new *Doom* games *Final Doom* and *Ultimate Doom*. The market not only became flooded with derivative FPSs, but also with almost as many *Doom* products.

In order to make their own FPS, several companies attempted to license id’s *Doom* engine instead of making one themselves from scratch (this trend continues to be strong today). One such game developer was Raven, their idea, *Heretic*. For *Heretic*, Raven used the same engine as *Doom*, but the look was different. The level designs were nicely done for the most part, despite the lower resolution texture for both the enemies
and structures. In *Heretic* the enemies were 2D animated sprites like *Wolfenstein 3D* instead of *Doom*’s digitized models. With the Doom engine it wasn’t actually possible for paths to cross, there was no room over room architecture, but for many gamers it was hard to tell that from the level design in Raven’s game. *Heretic* also used height to a much greater degree, the medieval fantasy setting called for high towers and large castles. They even included a power-up that allowed characters to fly. The game also had the ability to look up or down in the game instead of *Doom*’s static straight forward camera viewpoint. While not a huge impact on gameplay, it added more of a 3D feel to the game and a little more precise aiming for enemies at a different height level although the game still auto-aimed.

*Heretic*’s different approach to setting also brought in some improved features like an inventory system. This inventory allowed the gamer to collect and store items for later use. It was a major advantage in regards to health (health is based on a 100 point system, 0 is death 100 is full life. Health items increase the point totals prolonging the game character’s life) items because the player could use them whenever they were needed and didn’t have to do a lot of backtracking in a level to find unused health packs. It also allowed the player to store power-ups to use at their discretion. Maybe where the invisibility or invincibility sphere was picked up wasn’t a good place to use it, now the player could wait and use them when in dire need, such as when they neared death or found themselves outnumbered or ambushed. The most memorable item was probably the Morph Ovum which when used turned any enemy in its path into a chicken. The weapons in *Heretic* fit the medieval fantasy setting well but still feel like they are fulfilling an archetype. The crossbow felt like a shotgun with its slow rate of fire and
widespread trajectory, a gauntlet that fired magical bursts at a rapid pace resembled a
machine gun, and the phoenix rod was nothing more than a glorified rocket launcher
firing fireballs instead of rockets. None of the changes were too significant or creative of
a change to break the weapon archetypes that most FPS games employ. As far as the
game’s storyline goes, if a gamer didn’t read the back of the box or the text screens he
would have no idea what he was trying to accomplish and why. It had a straight ‘find the
key and get to the exit’ style gameplay. In the early years of the FPS story was not
central to the success of a FPS game. The story was just thrown together to get the
setting of the game to make sense, and sometimes even that didn’t work.

LucasArts released Dark Forces in 1995 and began a series that is still continued
today even though it strayed from the pure FPS track. The game (based in the Star Wars
universe) was one of the first to break many of the standard FPS gameplay styles and
archetypes. LucasArts developed the game engine themselves and it had quite a few
improvements. The engine not only allowed gamers to look up and down instead of a
static straight forward view, it also added the ability to crouch and jump. These logical
improvements added a lot to the strategy of the gameplay as well as giving the level
designers more creative freedom to build the game’s environments. In a firefight players
could jump for cover and duck behind obstacles to avoid lasers. Plus instead of just
walking from room to room, players might have to crawl through an airduct or jump
across an abyss to reach their destination. These features gave the game a much more
cinematic action feel and the levels were consequently more engaging. Overall, the look
of the game’s environments was well textured and fit very well with the atmosphere
established by the Star Wars films. Imperial Bases and Detention Centers looked
identical to the movies as well as new locales that made sense with the theme of Star Wars. Gamers never had to suspend disbelief of the setting because all the levels sufficiently represented the established Star Wars universe. Also, the 2D drawn and animated sprites of the Imperial Army and the various alien species smacked of authenticity. This authenticity was slightly overused when LucasArts decided to make every officer and alien species look exactly alike. The most common enemy in most of the levels, the StormTrooper, never changed through the duration of the game. The weapons in the game fell into a few archetypal patterns, but were all cleverly designed and fit into the game’s continuum. All of the weapons had a secondary firing mode that was unique to the individual weapon. They were more powerful than the primary firing technique, but with greater energy consumption. The player had to know every weapon’s modes and firing capabilities and when best to use each one. Thus the game had a tad more strategic appeal to it than its contemporaries.

One of the most differentiating elements that separated Dark Forces from the other FPSs of its day was its story. This game actually had a story! It was very well written and designed. Very little of the game’s plot was delivered through text. The only text appeared as a detailed mission briefing outlining your objectives for the mission. The rest of the story was given in animated cut-scenes, both rendered by the engine and by CG movies. The gamer also learned what to do next through conversations between Kyle Katarn, the main character, and the other characters he encountered. Dark Forces broke the ‘find the key and get to the exit’ gameplay. The levels had long and complex objectives from capturing a crime lord to planting detonators at key points in an installation. In some missions the objectives changed due to circumstances outside the
player’s control. *Dark Forces* was more than a well designed action game, it was an interactive novel.

Another noteworthy FPS released in 1996 was Raven’s *Hexen*. *Hexen* is actually the sequel to Raven’s other FPS *Heretic*. For this sequel Raven used the same enhanced Doom engine, but put a lot more effort into the visual and technical design. They may have used the same engine, though at first glance it doesn’t appear so. There is so much more detail and atmosphere in the game. Leaves blow around and fall off of trees, thunder bellows, lightning flashes in the sky, and lava boils. The game’s levels also have a lot more atmosphere and ambiance to them. The player can hear wind blowing, owls hooting, water flowing, water dripping in caves, and frogs croaking in swamps. Each environment is brought to life through these sounds and plunge the player deeper into the game. Structures in the environment can also be destroyed. To find items one can smash a vase or window to see if anything is inside. This helps add a lot of credibility to the game’s environment. Apart from being able to jump and interact with the surroundings, the levels can also interact with the player. Raven threw in a variety of well timed and adrenaline rushing traps. Crushing ceilings were standard from the days of *Doom*, but *Hexen* has collapsing floors, spikes coming out of the floor, cracking ice, and fire shooting walls. The monsters aren’t the only thing keeping the player on alert, these environments are themselves treacherous.

The monsters in *Hexen* were also given a lot more time and thought compared to some of the bland and simple enemies in *Heretic*. While a few *Heretic* monsters crossed over into *Hexen*, most are very well drawn and animated. They complimented the atmosphere of the game with an eerie edge. The Dark Bishops are a prime example.
These cloaked enemies float in the air and toss green fireballs at the player while dodging the player by fading in and out of the player’s view. One can’t kill what they can’t hit. The player always knows when a Dark Bishop is in the area because they chant dark tones constantly which can be heard before they are seen. The Centaurs are another good example of a well designed monster. The shields they carry aren’t cosmetic. They can block the player’s attacks readily, and often do so. The player has to be quick and strategic to hit them clean.

*Hexen* also has an interesting character system. The characters are still medieval, but the player can chose to play as one of three classes (Fighter, Cleric, and Mage) and the gameplay differs substantially depending on the initial selection. The Fighter is more powerful and quick, while the Mage is able to cast devastating spells. The Cleric is a mixture of both skill sets each in a smaller measure. The game has twelve weapons, but they are specialized by class, each class having four weapons unique to them. All classes of character still able to collect the same types of items for the returning inventory system (though the Morph Ovum is replaced by the Porkalator). *Hexen* is a pretty long game, so one might complain that each class has only four weapons to choose from for the entire game, but each weapon is at least cleverly designed and stays away from the standard arsenal of weapons the other FPSs overuse.

The narrative in *Hexen* may not be very strong, as it is a standard “stop the bad guy and his minions from overrunning the heroes’ realm”, but the level system is innovative compared to other shooters of the time. Every FPS at the time had linear gameplay. The gamer started at the beginning of the level and worked through it to the end. There was very little backtracking and once an exit was reached the level was over
and never revisited for the rest of the game. *Hexen* did things differently. The game was divided into several hubs, these hubs allow you to travel between six levels in a non linear fashion. The player has to hit a switch on one level to open up an area on another level. This allows for each of the levels to be explored multiple times in order to open the gate to the next hub. To keep things interesting, repeat exploration opens up more areas with monsters and sometimes they respawn (the game randomly generates a new monster) as well, keeping the player on his/her toes. It wouldn’t be as enthralling if a player cleared out a level then was able to run back through it again with no opposition.

The final noteworthy game in the first series of *Doom* clones was released in 1996 by 3D Realms: the classic *Duke Nukem 3D*. In the early years of the series *Duke Nukem* was a 2D side-scroller but for this installment the developers entered FPS genre. And that’s exactly the best way to describe *Duke Nukem 3D*. Incredibly Stylish. 3D Realms used their own proprietary game engine which it looked fantastic in its day. The environment was very modern and the cities and indoor sections (as well as the futuristic alien space stations) were produced with unmatched detail and high resolution textures. The game gave the joy of marveling at architecture while blowing it up. *Duke Nukem* has immensely interactive levels which compliment the tone of the game as described by Tim Soete of Gamespot:

“Duke allows you to be an action hero to the tenth power. If it moves, shoot it. If it doesn't move, shoot it. Anything and everything can be destroyed. Coin stuck in the soda machine? Pipe bomb the sucker. Toilet won't flush? Nothing a few rounds from your doubled-barreled shotgun can't handle. You slay your way through each scenario, filling aliens full of lead, even shooting wounded
adversaries as they bleed on the ground, whimpering for mercy. And all the while, Duke punctuates the violence with the type of flippantly cool asides required of our modern-day action heroes: "It's time to kick ass and chew bubble gum, and I'm all outta gum."

Few games up until this point had as much charisma and humor as *Duke Nukem 3D*. What little plot the game had, which was based around aliens stealing women from earth, was more than made up for by the humor. The aliens were as well designed and creative as the weapons. Although they were the standard 2D sprites, they looked better than any contemporary sprite graphics. And Duke could now duck, climb, jump, fly, and swim to dispatch them. The aliens could duck and fly to avoid Duke’s shots as well, which added a great deal to the firefights which were already top notch. *Duke Nukem*’s arsenal had the standard weapons: the shotgun, machine gun, pistol, and rocket launcher, but also a few really creative additions. One was a shrink ray that reduced almost any alien (though some of the tough large ones were immune) to a tiny size so Duke could step on them. The freeze gun turned enemies into ice statues which promptly exploded in the mayhem (and the freeze gun blasts bounced off walls which allowed the player to freeze around corners as well as himself accidentally). Perhaps the most ingenious weapon was the pipe bomb. These bombs could be tossed through windows, doors, and around corners and detonated when the player wanted. The gamer could also lay several down and set traps for enemies.

If any game earned a mature (M) rating, it was *Duke Nukem 3D*. Blood would splatter on walls, bodies would explode into messy gibs (bloody chunks of flesh), and aliens would choke and cough up blood before dying. But that wasn’t the only reason for the rating.
As mentioned *Duke Nukem 3D*’s plot revolved around an alien race stealing the earth’s women. Many of these women would be encountered in cocoons and were often destroyed in the crazy firefights. However those weren’t the only women represented. Duke’s travels sent him through many strip clubs and bars where scantily clad women were dancing. Walk up to one and press a key and they would flash you as Duke passed them a hundred dollar bill while saying “shake it, baby”. Many of the urban levels were strewn with advertisements for porn films and sex shows. At first that it sounds pretty tasteless, but after playing it one can see it was done in a tongue-in-cheek style. It was a great satirical work echoing of many classic action films and themes. The game even came with a feature that allowed parents to turn off the mature content and password protect it so households with children could prevent viewing and play. For years many FPS titles tried to take the crown away from *Doom*. *Duke Nukem 3D* was a strong contender to take the prize with its great action, levels, and witty humor.
Chapter 4

Established Dominance

One way to describe id’s 1996 game, *Quake*, is as a modern version of its previous game *Doom*. The plot was the same, the variety of weapons was almost identical, and the enemies, while looking different, behaved the same. So why then was *Quake* so talked about and considered so revolutionary that it generated so much buzz when it was released? Simple, it was the first truly three dimensional (3D) game (*Kushner* 204). *Quake* was the next important step toward the goal of creating a feeling of ultimate immersion. For the Quake engine, Carmack had to start completely from scratch; none of his former graphics engines could be implemented in any way. FPS games in the past only simulated 3D, the player had a limited field of vision and movement while in-game characters were 2D drawn sprites. Though many of these games were great in their own regards and brought different elements to the table, *Quake* was the FPS to push the genre into a new evolutionary stage.

The first noteworthy advantage of a true 3D engine is that the game architecture becomes true to life. Rooms could be placed on top of rooms to create highly complex buildings and structures. With this design concept at hand, the designers created ledges, holes in the floors and ceilings, and nooks and crannies to house useful items or shelter foes. If a player could see a location, there were almost always multiple ways to reach it. With realistic buildings came more believable physics models. Physic’s effects and particles are created in real-time or in reaction to a players actions, instead of being pre-drawn. When a rocket hits a wall the explosion expands sending particles everywhere instead of being depicted as a flat 2D fire ball. The combination of these elements allows
the player to adapt new strategies previously not possible. The grenade launcher can be used to accurately ricochet grenades off of walls and other structures to bounce behind doors and around corridors.

Along with the environment being true 3D, so are the characters in the game. Instead of using 2D sprites, the enemies were created with polygons because hundreds to thousands of these polygons can be pieced together to create three dimensional beings. Like the rest of the game content, 3D models can be viewed from any angle as if the player was actually walking through the level. These new models are also represented in high resolution. At the time *Quake* was released, the monsters looked light years ahead of the competition. This new level of realism greatly enhanced the gameplay. When shot, the enemies would fly back hitting walls or drop to their knees from the impact allowing the player to fire again before they got back up. Gibs were no longer canned animation where the body parts would fly the same way every time. If hit by an explosive weapon, the dismembered parts could hit and bounce off of any structure in any direction. This was an unmatched level of realism and set the benchmark for games to follow.

The evolutionary 3D environment also presented a brand new element of control. FPSs that preceded *Quake* were all played with the keyboard. The player had an entirely fixed point of view even though some games had widened that view a little by allowing the gamer to look up or down to a certain extent. Auto-aiming was used for enemies that couldn’t be sighted directly down the barrel of the gun, just aim in the general direction and the game did the rest. *Quake* introduced the controls that would forever be engrained into FPS gaming on the PC, the mouse and keyboard combination. The keyboard is used
for basic up, down, left, and right movement and other variety of commands, while the mouse was used to aim and fire. It was like moving through the game world though the eye of a camera. The days of imprecise aiming were over. The gamer had to place the crosshairs directly over what they wanted to shoot at. This method of control could quite possibly be the greatest addition to FPS gaming, no other formula has ever matched the speed, accuracy, and fluidity of this combination. Today many gamers refuse to play FPSs on console systems because the joystick controls on the controllers aren’t even close to the speed of the mouse.

On the multiplayer side of the coin, *Quake* introduced a new form of internet playing. With *Doom* and the games after it, players had to dial directly to one another in order to play a game or connect the PCs directly over a LAN, which only allowed players connected to the same network to play. With *Quake*, many people had setup servers dedicated to hosting *Quake* games. Up to sixteen people could connect to a particular server from wherever they were in the world and deathmatch with each other. Soon after *Quake*’s release, hundreds of servers were up hosting games and players would surf server lists looking for games to jump into. Quake also popularized tournament gaming and the forming of Internet clans (term in gaming used for teams of players). Tournaments were held all over the country and players competed not only for prestige, but also for cash and prizes. In local areas, groups of gamers would get together and form clans and battle for positions within them. Tournaments would also be held where clans from nearby regions could deathmatch against each other for top rankings in the clan nation. What began as a game genre targeting single players was rapidly become a highly social and close knit community.
Upon first glance, *Quake II* was hailed as the next great advancement in FPS gaming. It was everything that *Quake* was with higher tech graphics and support for more Internet players. In reality, *Quake II* was not as ahead of the competition as everyone had initially believed. It was basically the same ‘run and gun and find the exit’ style gameplay with no story and dimwitted Artificial Intelligence (AI) that could now duck. However there were a few things about *Quake II* that did advance it beyond FPSs at the time.

*Quake II* utilized the growing market of graphic accelerator cards for PCs that were emerging before and after the game’s late 1997 release. These 3Dfx cards were designed to handle complex graphical programs and take the strain off the PC’s processor. With the processor free to handle other computations, the graphics card could push visuals to the limit without sacrificing system performance. Since not everyone was able to buy these new cards because of the initially high costs, *Quake II* gave the option to run off of either the hardware or software acceleration. A player could still experience the game using software rendering, but those capable of using hardware acceleration would get enhanced graphical features that would make the difference almost as drastic as night and day. The 3Dfx cards allowed *Quake II* to run in a high resolution mode called OpenGL which gave the game incredible new colored lighting (realistic transition gradients between light and dark areas), smoother highly detailed textures, and more fluid play (Kushner p237). The days of blocky, pixilated textures were numbered and anti-aliasing (blending of textures to remove jagged and blocky edges) had arrived.

Another important feature in *Quake II* was the implementation of weapon balance. FPSs all had a decent variety of weaponry (while cliché at times) for the player
to choose from, but the weapons weren’t very realistic in their execution. Some were too weak, others too powerful with an unbelievable rate of fire. *Quake II* started to change that a little. The two shotgun types had different rates of fire and varying kickbacks allowing the player to be more careful and strategic in their use. The rocket launcher was tweaked to lower the damage and it took longer to fire and reload. In the days of *Doom* a gamer could lob rockets as fast as shooting bullets from a pistol. The change helped balance the multiplayer side of *Quake II*; no longer could players who found the rocket launcher first dominate a game. Machine guns gave more a kick and the high caliber minigun took a few seconds to warm up before firing and cool down. Perhaps the most interesting weapon was the railgun, which fired a high powered uranium slug with deadly pinpoint accuracy. To balance the guns immense power and insane sniping range, id gave it a long reload time. The player would have to lead (aim in front of a moving target to hit it) his/her target very carefully or be left wide open for a counterattack. In many FPSs at the time, the weapon the player chose to use would be placed dead center in the screen. For many guns, it is impossible to hold them in this way because of size and power constraints. The *Quake II* designers placed the gun in the now standard lower right side of the screen (left oriented people could choose to change it to the lower left side). The balancing of weapons not only makes the single-player mode more strategic, but it also helps make the multiplayer side more enthralling and less likely for one person with a big gun to be able to rule a game map unchallenged.
Chapter 5
Unreal Competition

The year 1998 sparked a change in the FPS community. Until this point almost every revolutionary game that pushed the FPS higher on the evolutionary ladder came from id Software. Epic MegaGames stole a hefty portion of the spotlight when they released their own FPS, Unreal, created with a different graphics engine which boasted a completely unique look and feel. Unreal had big task before it, Quake II had been out for several months with a much bigger fan base behind it. However, Epic was confident that their game's unique new style would appeal and top Quake monster.

At immediate glance, everyone could see the most obvious difference between Unreal and Quake II. Unreal had amazingly colorful textures and environments, while Quake II was over-shaded with dull browns and greens. The 3D technology went a long way, but it was far from looking like a seamless world. The environments of Unreal were teeming with life and detail. Water was perhaps the best feature with reflective and clear properties as well as swimming fish. Unreal skimps on the story a bit, the basic premise is the player was prisoner on a ship that crashed on an alien planet. As the player roams the planet he/she finds one race of low-tech aliens, Nali, are slaves to an advanced race, Skaarji. By helping free the Nali from their oppressors, they help the player find a way to escape. The variety of textures and palettes totally distinguish one setting from another. The player never mistook Nali Temples for Skaarji ships. Unlike many other games around at this time, Unreal offered a mix of open outdoors and cramped indoor levels. Each setting was also textured and detailed to give them different looks keeping
the player from getting bored with heavily overused rooms. It was a very refreshing change of pace.

Unfortunately, the varied environments didn’t transfer into the enemy design. *Unreal* only had six types of enemies, even though Epic tried to spice them up by creating several different color patterns for them. After only a few hours of playing, and the game was quite long for an FPS at the time, the player had seen every enemy that the game would throw at them. Another compensation for the lack of enemy designs, Epic made the AI pretty intelligent, especially the Skaarji. The Skaarji could effectively dodge the player’s shots and shoot back while the player is busy reloading. These aliens could also power-up a personal shield if they felt it was needed (which was usually all the time, if they had shields). Smart AI makes for some really exciting and intense combat, which is the core gameplay of any FPS.

Since combat is the core of the FPS, in order for it to be successful good AI is only half of the equation. The other half are the weapons made available to the gamer. The mixture of medieval and sci-fi architecture wasn’t the only interesting blend in *Unreal*. Instead of using basic shotguns, machine guns, etc, Epic created some weapons the likes of which gamers hadn’t seen before. The Flak Cannon is one that probably stands out the most. One could probably best compare it to a shotgun, it fired shells of red hot metal chunks that spread like shotgun pellets but unlike buckshot the flak could ricochet around corners. The Flak Cannon also had a secondary firing option which lobbed a shell like a grenade that exploded on impact. Employing such unique weapons posed interesting problems for players. With no prior examples, gamers had to adjust their playing style to the artillery. They had to learn the guns strengths and weaknesses
from scratch and then decide when to use them all while trying to make it through the game. While it was a genuinely unique problem, it also became frustrating at times.

With so many interesting developments, *Unreal* wasn’t without its problems. Aside from the unusual weapons and small adversary variety, the combat could get repetitive. The game was quite long for its time and with little story to go on, the compelling reason to play came from discovering new environments and experiencing the firefights. But with a long game, the fights with the same types of enemies and same weapons, although different, became hackneyed in its own right. Also the game shipped with a multiplayer segment, but it was next to impossible to find an internet server for a long time after its release. Its single player remained a big draw despite a few problems and provided more than enough competition for the *Quake* series.
In mid 1998 Epic stunned the gaming world with *Unreal*, which proved that major evolution in the FPS genre could come from a company besides id. Having just experienced *Unreal*, the FPS gaming community was taken by storm with an unknown company called Valve Software. A group of ambitious designers set out to create an FPS that would put all others to shame. Needless to say, the gaming industry enthusiastically subscribed to Valve’s promises and hyped the game as much as the next *Quake* or *Doom*. Too much hype can be counterproductive and having been fooled before, many Gamers expected the title to flop. *Half-Life* did everything but flop. In fact, it was hailed as the most revolutionary FPS since *Doom* and the greatest single player experience across all genres.

*Half-Life’s* success was due to several factors. For starters, Valve used a compilation of engines to create the game environment. Valve licensed the Quake and Quake II engines as well as making several of their own modifications to the code. The result was the most seamless and realistic setting ever seen. It exceeded the drab featureless landscapes as *Quake II* and was more detailed, but more realistically colored (no unusually bright palettes) than *Unreal*. The game’s story takes place in a secret government facility called Black Mesa and the engine renders it beautifully. Like *Unreal*, the environments are so well textured and varied that the player never encounters the same area twice unless they have to backtrack. Valve took great care in creating a hyper-realistic world and keeping it consistent throughout the entire game.
The story isn’t much more fleshed out than other games in the genre, but its how its told and presented. There are no cut-scenes or text screens explaining objectives or what to do next. The player is told what to do next by interacting with NPCs (non-player characters). There are also no levels. The entire game is built up of sections that encompass the entire facility and the surrounding area. The only break in gameplay was the small loading times in between these sections. The story is told as the game progresses like a film. The player is Gordan Freeman, a theoretical physicist employed at Black Mesa. He, along with the other scientists, perform ground breaking experiments. The game begins with Freeman riding the train to his scheduled experiment. During the experiment, in which the player takes part, something goes horribly wrong and suddenly aliens begin warping in from another dimension all over the facility. It becomes a fight for survival and escape while trying to push the aliens back. The scientists not only have to worry about aliens, the government has sent in troops whose idea of cleanup is annihilating everything that moves.

All the character models are as detailed as the environments. The aliens border on the line between beautiful and terrifying. The headcrabs are probably the enemy that stands out the most. They’re a cross between the facehugger from Alien and a crab which after they attach to a head, control the body and twist it into a dangerous monstrosity. These little things look slow, but are pretty cunning and can lunge across rooms in a single leap. The worst part is that they make little noise, so the player might not see them until they hear the scream and the headcrab fills the screen. While the player encounters the same enemy types quite a bit, there are so many that it never really gets boring. And the game saves many surprises for later in the game. The only modeling complaint is that
the troops all look alike. Valve should have created more textures for the human
enemies.

What the marines lack in variety, they more than make up for in intelligence.
Valve promised the most advanced AI in any game to date, and they delivered on that
seemingly impossible task. The aliens were fairly intelligent; they could chase the player
through doors and retreat if they were taking too much damage, but paled in comparison
to the marines. The Marines were aware of their environment. In many games at this
time, a player could shoot sniper style at an enemy and others around him would ignore
the fact that a comrade had just fallen dead. Not so in *Half-Life*. One sound from a
weapon and the marines were at alert and behind cover firing at the player. They ducked
under crates, ran behind walls, and feinted. If the player was foolish enough to stay in
one spot or get cornered, one marine would toss a grenade to flush the Gamer out while
his comrades were waiting with MP5s locked and loaded. Rushing Freeman was also a
favorite tactic. In many games one could run behind a wall and wait for an adversary to
emerge and gun them down. The marines were smarter than that. Most of the time they
waited for the player to pop out and if they did play cat and mouse they played with guns
blazing or worked to flank Freeman. *Half-Life* was a dangerous place. If the
environment and aliens weren’t trying to kill Freeman, the marines were.

The realism didn’t stop with the AI and the environments, Valve went all
the way. *Half-Life* has no power-ups. No invincibility, no invisibility, and no quad
damage. Freeman is wearing a hazard suit, and to keep the suit at full power he has to
charge up at power stations or find small batteries. That explains armor. Health is also
gathered at stations and in packs on shelves and packed in crates. The player will never
find anything arbitrarily floating in thin air like a gift from God. To collect ammo
Freeman must find caches around the facility and pick it up off of dead opponents. The
weapon set has its usual set of pistols, shotguns, and machine guns. The difference is that
all of them are built off of real models. Rather than elaborate on a gun, Valve made them
as realistic as possible. The rocket launcher can only hold one rocket at a time and
Freeman has to manually reload them adding to the rate of fire. The MP5 comes
equipped with a grenade launcher like its life counterpart, and the shotgun can hold 8
shots until Freeman has to reload all 8 one by one until the chamber is fully loaded.
Being that the game takes place at a research facility, the sci/fi weapons all fit in without
coming off as too outlandish.

Valve Software has obviously spent a lot of time studying the mistakes of
the past. The result is *Half-Life*, the closest thing to a revolutionary step the genre
has ever taken. Through a series of subtle and artistic design decisions, *Half-Life*
creates a reality that is self-contained, believable, and thoroughly engaging. And
while it may be surprising that no game has utilized any of these ideas in the past,
it's clear that any future shooter will be remiss to overlook them. (Ron Dulin,
Gamespot)

*Half-Life* pushed the FPS into the era of cinematic storytelling.
Chapter 7
Multiplayer Mayhem

After *Quake II*, id came up with an interesting and bold new idea for a game. Over the years, the multiplayer side of id’s games became more popular than the single player experience and is why people kept playing their games for several years. Id’s next release, *Quake III Arena*, was a straight multiplayer game only. Instead of spending all the development time creating a single player component when most people wouldn’t even finish it, id decided to give players what they wanted. *Quake III* had a single player section, but it was more of a simulated multiplayer. Basically, the player picked a character and ran through several tiers of maps defeating computer controlled opponents called “bots”. These bots were supposed to simulate actual players so the person playing could practice a bit before jumping right into an online game. However the bots weren’t the most intelligent of opponents. A player might have an easy or hard time dispatching them depending on the difficulty setting they chose, not the cleverness of the bots. The idea behind *Quake III* is simple, jump into a map, pick up a gun or two, and try and kill as many opponents as you can. That’s it, pure deathmatch. The multiplayer side is a bit more interesting since there are live players, but the strategy is pretty much the same as the single player version. *Quake III* has a few more modes of play like team deathmatch and CTF (capture the flag), but the only real difference is that there are fewer targets to shoot.

The real genius behind *Quake III Arena* is the graphics engine Carmack created. It was the most technologically advanced and prettiest engine out at the time. The maps were highly detailed and textured complete with volumetric smoke and realistic fogging.
effects. The levels were beautifully rendered and designed with deathmatch solely in mind. The great designs didn’t stop with the architecture; the character models look amazingly well textured and animated. Most of the characters (as well as the maps) were designed from id’s *Doom* and *Quake* worlds creating a great mix of the two games. The weapon set sounds as good as it looks. *Quake III* has a good mix of shotguns, machine guns, rocket launchers, the sci/fi railgun, and the returning BFG from the original *Doom*. All are well tweaked so no one weapon can rule a map. The *Quake III* graphics engine produced such breathtaking results that game developers have been licensing and using it since 1999. Many of 2003’s game of the year contenders were made using the *Quake III Arena* engine.

Taking a cue from id’s announcement of *Quake III Arena*, Epic MegaGames, creator of *Unreal*, decided to release a multiplayer only version of their FPS. The aptly named, *Unreal Tournament* was released only a week before *Quake III Arena*, no doubt for direct competition with id (Kushner p276-77). *Unreal Tournament* ran directly off the same engine as Epic’s *Unreal*. Same slick style textures, colorful graphics, and brilliant level design. Seeking to right the wrongs of *Unreal*’s multiplayer aspect, which took months of tweaking after its release to make it more stable, an *Unreal Tournament* player could easily find a server to play on right out of the box. Once in a game up to and over 15 players could battle across a huge plethora of maps that felt more like real places rather than *Quake III*’s outlandish, dreamy maps.

Like *Quake III Arena*, *Unreal Tournament*’s main draw was deathmatch play, but unlike its rival, Epic offered a total of six modes of play all of which were superbly implemented. They included Deathmatch, Team Deathmatch, CTF, Domination,
Assault, and Last Man Standing. Domination and Assault were two modes totally new to the multiplayer. In Domination two teams fight to take control of three key points on a map. To take a point, a team member simply ran over it and the team scores. The longer a team holds the points the more the score goes up until the limit is reached and the map is dominated. Assault also involves two teams; one team is defending an area while the other team is attempting to take it. The assault team has a certain amount of time to take the area, and if they do take it the roles are switched and the assaulter becomes the defender. These six exciting modes of play pushed *Unreal Tournament* over the edge and added tremendous life to its replay value, especially with mapmakers pumping out new levels on a regular basis.

Great maps and multiplaying options weren’t the only component of *Unreal Tournament* that turned players on. Epic designed a set of weaponry innately better than they previously had for *Unreal*. The more ingenious weapons remained, like the Flak Cannon and Shock Rifle, but they also included a better rocket launcher, sniper rifle, minigun, and the ultimate power weapon, the Redeemer. Every weapon had a primary and a secondary firing mode, essentially doubling the weapon count. Not only did this make the battle more intense but also added strategic elements to the battle. A player almost about to die could switch to the minigun and use its secondary option to fire bullets at a much faster rate as a last ditch survival effort. By holding the fire button down, the rocket launcher could load up to six missiles and fire them all at once adding much more punch and a wider blast radius. A large spread is very useful for those hard to find snipers. *Unreal Tournament*’s most powerful gun, the redeemer is basically a portable nuke. It only had one round per use, but the player could fire it in a straight line
or control it through a guided missile camera. In an ode to the balance of the weapon set, no player was sure to win with one particular weapon. The redeemer’s missile could be shot out of the air before it reaches its destination.

If all these advancements over *Quake III Arena* weren’t enough, Epic also had something in store for players who from time to time were forced to play alone. The bots in *Unreal Tournament* were not only scalable in difficulty, but actually behaved like online human players. At higher settings, the single player deathmatches were just as exciting and challenging as the multiplayer. Adding more spice, *Unreal Tournament* was extremely customizable. When setting up bots the player could control bot intelligence individually, as well as a favorite weapon, how much damage it does, and how fast the players move. Map setup also gave the player an option to use ‘mutators’. Mutators set special rules for the match, from chainsaw melee, snipers rifles only, to ‘Fatboy’ (players get bigger as their frags (kills) increase). With modmakers pumping out more maps, new mutators, and tons of new character models to use in addition to the already vast character creation, *Unreal Tournament* was considered by many to the winner of the multiplayer battle with *Quake III Arena*.

A discussion about multiplaying in late 1999 could not be complete without mentioning the king of popular muliplayers. The mod community surrounding *Half-Life* was huge, and out of this community came the mod called *Counter-Strike*. When it was first created, *Counter-Strike* was available as a free download only requiring the Gamer to have the full version of *Half-Life*. After watching this mod grow rapidly in popularity, Valve’s publisher decided to release a stand alone retail version of *Counter-Strike*. *Counter-Strike*’s immense following is a big of an anomaly at first look. Its not as
graphically brilliant as *Quake III Arena* nor does it offer even a third of the options of *Unreal Tournament*. *Counter-Strike*’s following comes from its basis on realism and team play.

Working as a team is critical to success in *Counter-Strike*. A player picks to play on the terrorist side (trying to plant a bomb, assassinate a VIP, or taking hostages) or on the counter terrorist side (attemping to thwart the terrorists). The game ends when the task is completed or when one side’s team members are completely annihilated (which is usually what ends up happening). No one person on a team can accomplish this. In fact, when a player is killed they are out for the entire round and watch the rest of the round play out through a ‘free floating’ camera. Since there is no continuous respawning of players typical in other deathmatch games, players need to work together as a team to win the match. To place further emphasis on teams, more money is received (used to buy weapons before the round starts) for winning a round than for killing opposing players.

Speaking of weapons, all the arms in *Counter-Strike* are based off of real world counterparts in appearance and sound. Each gun has its own strengths and weaknesses that the player has to learn. *Counter-Strike*’s environment is very well suited for learning playing styles, which is essential to playing well (it takes a lot of practice and patience to move up the ranks among all the veterans). By watching the veteran players by means of the free floating camera, new players can see how various players play and use the weapons available. This learning opportunity is part of the huge draw to the game; people who’ve been playing it for years are still learning new techniques from other players. The *Counter-Strike* community is the most played, widespread, and diverse game community
in the industry. Released in 1999, *Counter-Strike* is still the most played game on the internet and a version was just released for Xbox to attract players to Xbox Live.
In the late 1990s the FPS genre had been around for only about seven years and it already seemed that the genre was becoming stagnant. While many game developers were releasing some ambitious and complex titles, they became overshadowed by media hyped games that were only demonstrations of the growing technical power of the personal computer. id and Epic’s ideas of gameplay ran rampant in every FPS title released around them. Run and gun gameplay was relatively simple to slap together and then tack on a weak storyline to be sure the action in the game made sense. While *Half-Life* was a very noteworthy exception to this rule, games like *Unreal Tournament* ruled the marketplace. That’s what gamers wanted. In the next couple of years, a few developers were set on the notion that the industry hadn’t seen everything the FPS was capable of and set out to prove it.

Looking Glass’ answer to the now overflowing FPS genre completely took the gaming community by surprise, with a new direction and element of play. The release of *Thief* coined a new term for similar games that would follow, the “first person sneaker”. Unlike all the other games of its time, *Thief* relied almost exclusively on stealth play. The suspense and tense moments in the game didn’t come from running from room to room blasting everything that moved, instead it came from waiting for the right time to strike or to slip by unnoticed. In fact, it was entirely possible to run through *Thief* without killing a single person and in many ways that concept was consistently reinforced.
The game’s main character, Garrett, was a master Thief not a fighter or warrior. He’s not proficient at combat within the medieval fantasy setting in which the game takes place. Garrett’s use of his sword was only as a last ditch effort and it was hard to dispatch foes with the weapon. A player who jumped into a room while swinging his sword would not make it very far at all. The gameplay of Thief was mastered by sticking to the shadows and avoiding being spotted by enemies at all costs. To aid the player in accomplishing this, the bottom of the screen displayed a light meter that relayed how visible Garrett might be to the people he encountered. While the game was set during the night, in dark buildings, and cramped passages, it was no easy task to hide. Not every light source was a torch that could be strategically extinguished by a water arrow (which Garrett had in limited supply). On top of that, Thief is probably the first game in which sound was extremely important to success. Not only could the player use it to find and track guards, but it could be turned against them. Depending on the type of flooring, Garrett’s footsteps might be heard by foes. Running on carpet was no problem but a race across metal plated or tiled floors became a problem since every guard within earshot would hear it.

The AI in Thief is definitely advanced and character models move very realistically due to a motion capture technique which made for some seriously intense gameplay moments. Enemies are aware of their environments. If the player shot out a torch, characters would notice and inspect the area. Few things are more suspenseful than putting out a light and then having the man on patrol scour the vicinity for you, sometimes even walking right past your hiding spot. There were even a few cases where the guard would walk right towards your location, stop right in front of where you were
crouching, and look deeply into the darkness. The guard would be so close that had the
game allowed such interaction, you could reach out and touch his/her face. If the player
made one false move, the jig was up. If the player misjudges the arch of the arrow and it
missed a guard, he might run away, call for backup, or seek Garrett out once he heard the
clang. The solidity of the AI wasn’t perfect, but it allowed for so many different
possibilities in conjunction with the environment, that every player’s experience within
the game was different. The AI reacted to how the player chose to deal with them. Very
few games allowed excitement and replayability at that level.

To further draw a player into its seamless world, *Thief* had a great amount of
realism and artistic value. Every cutscene is animated and voiced professionally, setting
it miles above the competition at the time (and even above many games today). The
symbols and quotes employed throughout the game opening and level introductions had a
meaning and purpose the player would only distinguish as he moved through the story.
The plot is still one of the best today and home to perhaps the best twist ending ever
featured in an FPS.

Looking Glass spared no expense, not even in the design of the graphics engine
which they could have easily done since the game is pretty dark. No detail was hidden in
the darkness and it provided a stark and artistic contrast with the lighter parts of the
game. Looking Glass helped bring the FPS out of a rut and in a list of examples arguing
gaming as an art form, *Thief* is at the top.

Hiding in the dark corners of the id dominated FPS scene, a group of games set
out to do something unique among gaming by crossing genres. A short time after *Doom*,
Looking Glass released a game called *System Shock*. With this ground breaking title,
they combined role-playing (RPG) elements into an FPS game. Unfortunately, it never received much attention from the gaming mainstream. In 1999 Irrational Games (licensed and published by Looking Glass) released the sequel, *System Shock 2*. Just like its predecessor, *System Shock 2* combined RPG elements with an FPS formula. The result was nothing short of outstanding and made much more of an impact on the gaming industry than the original.

Unique to the FPS, *System Shock 2*’s first implementation of RPG elements was in the area of character creation. In RPG’s, before a character begins the adventure they player selects from classes, skills, and attributes to mold the game experience to their preferences. Through the game’s journey the character is allowed to improve or take on new skills based on the amount of experience points they’ve gained. *System Shock 2* gave players a choice between being a marine, a hacker, or an OSI (Psionics). Each of these classes presented the player with different skills and physical attributes with which to begin the game. Progressing through the game a player was rewarded with cyber modules either by completing objectives or finding them. These modules allowed the gamer to upgrade their skills and attributes or to learn new ones. While the character creation wasn’t perfect (to progress through the game the player was forced to dab into the other classes’ skills in order to be successful. The beginning characteristics were different and every player ended the game as a sort of handyman) it gave players a distinct experience not found anywhere else in the FPS genre.

RPG’s are also known for complex interfaces in which the player must run though characteristic menus, manage inventories, and read messages pertaining to the game’s plotline. *System Shock 2* had these information pieces, but integrated them in a very fluid
design. No menu was useless or hard to get to, which could be critical to a players’ survival in an FPS. Regular FPS players didn’t have too difficult a time adjusting to the change and it enhanced the gameplay. Another RPG feature was heavy character interaction. This was how players got their information about the world they’re in and what to do next. Such interactions were extremely story based. While *System Shock 2* had practically no person to person interaction, there were plenty of unseen characters who left emails and logs to read in order to obtain information. In classic RPG fashion the characters were quite well defined and fleshed out even though the player was only listening to their journals.

*System Shock 2*’s most prominent feature was its story and atmosphere, like any good RPG should have. The story started with the player’s character waking up from his cryotube (freezes an individual for long interstellar travel) and finding the ship he’s on, the research vessel Van Braun, vacant of human life and infested with an alien worm-like parasite. On top of that, the ship’s AI had gone haywire and a more intelligent and sinister AI is trying to take over. The Van Braun’s mission was to explore the vast reaches of space. What was found in the dark void creates one of the scariest atmospheres in gaming. The ship very realistically created what a research vessel might look like. It felt like people actually lived and worked inside. The only problem was they were all gone, killed or taken over by the parasite. A lot of the fear was based around the empty, lonely feeling the player got while playing.

Using a modified version of *Thief*’s engine gave the aliens the ability to see and hear and the player the ability to try and thwart those senses. Since weapons and ammunition were quite scarce on the Van Braun, tension was always extremely high. To
help with this problem, players could research alien specimens in order to find out what was the best offense against a particular strain of parasite and gain advantages against them. The annelid strains were also given varying degrees of intelligence. The higher evolved annelids would creep around carefully rather than the brutes who would charge relentlessly on sight. Not only would this add to the suspense of the game, but gave it more credibility that is important in science fiction stories. The game’s plot was a big factor in keeping the player involved. They wanted to know what happened and what exactly these annelid parasites were.

System Shock 2 combined two very distinct genres in a way that not only appealed to fans of that genre, but also showed that the FPS can have great action and a gripping story. It takes brilliance to completely creep a player out just by reading the journals of a crew member slowly being taken over by a parasite with each entry and compel the gamer further. Few games have players yelling at their friends to be quiet because the aliens might hear them. System Shock 2 did and that was with the lights on.

Irrational Games wasn’t the only company to attempt and succeed at combining RPG elements with FPS gameplay. In 2000, Ion Storm released their own unique combination masterpiece called Deus Ex. Deus Ex didn’t let a player choose between several character classes or create a suspenseful atmosphere, but what it did do was create a living, dark, futuristic world of vast epidemics, rivaling secret factions, and global government conspiracies. Some might compare it to the popular show The X-Files, only with much more depth and intricacies.

Warren Spector (also a major player in Thief and System Shock’s design teams) set out to create a game with something Gamers have never had to deal with before --
cause and effect. *Deus Ex* was all about giving the player choices and playing out uniquely to the consequences of those decisions. Every objective in the game can be achieved through multiple means, no matter how huge or small the objective. Say the player wants to go through a locked door. First he could try and pick it, if he had enough lock picks and it’s pickable. If the door wasn’t very sturdy, he could try and blow it up with a LAM (Light Attack Munitions). Other routes included searching or persuading individuals for the password or electronically bypassing it. And that’s just for a locked door. There were hundreds of other situations the player faced and could solve however they chose. Some are even plot related. Like any good RPG, *Deus Ex* had hundreds of character interactions (this game has more lines of dialogue most of today’s major films (Gamespy, interview with Warren Spector). The way the player acted and responded to those characters has an effect on how willing they would be to help or hinder. If the player bragged to his friend in the munitions department about how many terrorists he killed in the last mission, the munitions officer wouldn’t give him extra ammo and scold him for his disregard of human life. Many choices had a more grand effect on the plotline (the game has three endings that can be obtained by making different decisions) but the idea that there were consequences to every action the player made was engrained early in the game and was a major part of the game’s appeal.

Choices would be uninspiring if the game world wasn’t interesting or lively. *Deus Ex* effectively created a game world that felt alive. The game took the player to several cities across the globe seeking the cure to a mysterious disease epidemic called the Grey Death which was overtaking the world. Did the government make the disease to gain power and profit off the cure? Where do the Illuminati and Majestic 12 factions fit
in? Needless to say, it’s a very complicated plot with many twists and surprises that kept the player busy with over twenty hours of gameplay. Each of the locations were bristling with life. Citizens roamed the streets, people had conversations in bars, workers complained about their job, and dealers sold drugs to junkies (which being an elite government agent the player could get involved in). Everyone had something to say and it paid to talk to as many people you could. You never knew who had information for you or a useful item. The player could even read magazines and newspapers lying about to find out more about the world, and even some of the players previous high profile encounters. The game world felt like there was more to it that what the player saw- like it lived and breathed even when its wasn’t being played.

_Deus Ex_ had all the RPG elements that _System Shock 2_ had; an inventory to manage, skills to build on, and documents to read and check. Unlike _System Shock 2_ though, the player started out as a predefined character like any other FPS. JC Denton was a new recruit to UNACO, a government agency charged with dealing with terrorists. Denton’s claim to fame was the fact that he is biomechanically enhanced with upgradeable augmentations. The player could chose to enhance parts of Denton’s body with the augmentations when they are found. Skill points were also given for completing objectives and could be used to upgrade Denton’s skills. The player started out with a clean slate, but chose how Denton evolved every step of the way. The player could mold Denton to their style of play whether to become a stealthy hacker, a silent sniper, a walking tank, or a mix of all three. It was totally up to the player’s discretion. _Deus Ex_ was a very ambitious game for its time. It attempted to do so much at once that not everything panned out due to the use of the somewhat outdated _Unreal_ engine for
graphics and an interface that was cumbersome to navigate and often got in the way of immersion. Despite these faults it is considered a revolutionary title not only for the FPS genre but also for video games in general.
Chapter 9
Rise of the Console FPS

For the first several years of its existence, the FPS remained solely a PC genre. The console systems out at the time were not advanced enough to run FPS games. There were a few ports (designed for PC, but reworked to play on a console) that made it onto the Super Nintendo such as *Wolfenstein* (the dogs were changed to rats and some of the Nazi symbols were removed) and *Doom*, but they never lived up to their PC counterparts. Console systems had problematically limited hardware resources and a small control scheme to work with. The first system to present an FPS in a successful manor was the Nintendo 64 (N64). Not only did it have the hardware to pull it off, but the controllers had a more precise method of control due to an analog joystick (early controllers only had a directional pad with allowed only four directions designed for side-scrollers and 3rd person games).

The first major console FPS on the N64 was Rare’s classic shooter *Goldeneye*, based off the James Bond film that came out earlier in the year 1997. Utilizing the N64’s high bit processor, *Goldeneye* had exceptionally great graphics especially since Nintendo decided to stick with cartridges for their games instead of moving to CD-ROMS. Many stated the games for the system had a blur effect to them due to the N64’s limited capabilities, but few let the distortion get in the way of fun. The game’s graphical design is shaped around polygons, creating 3D environments and enemies. While the game had a decent amount of detail, it was easy to tell at times that many objects were polygonal. The realism however was unmatched on any console FPS, and even many PC games at the time. The sound effects and music were hyper-realistic and the enemy designs were tremendous. The soldiers actually looked and moved like living people. The animations
were some of the best ever seen for any game. They would shuffle around, swat flies, and die in different ways depending on where they were hit and with what. The AI was also pretty smart and would duck and fire around cover as well as try and jump out of the way of explosives.

The huge part of *Goldeneye*, as in many FPSs, was its multiplayer option. The N64 console had 4 controller ports enabling four people to play against each other at once on one TV. The screen was split into four sections, but the format barely detracted from the play unless the TV was a very small size. For the first time in FPS gaming several players could not only play in the same room at the same time, but also with one focal point without having to lug their PCs to each other’s houses. Not only did *Goldeneye*’s multiplayer version have a decent amount of maps, but it also had a nice variety of weapons and a huge selection of options to play with. Players could take off radars, give guns paintball ammo, give their characters giant heads, add dual weapons and a slew of other options. These were all added benefits of completing hidden goals throughout the game’s single player mode giving both side tremendous replay value on top of the already phenomenal experience. Rare’s *Goldeneye* remained one of the top N64 games throughout its life and paved the way for the FPS in the console market.

In that same year Acclaim released *Turok: Dinosaur Hunter* for N64, also an FPS. While the setting was a bit different, futuristic and prehistoric at the same time, and the gameplay was similar to *Goldeneye* it stood up on its own and offered a different experience. The graphics were just as slick as its competitor, in fact maybe a little better. While *Goldeneye* had a grainy blur attached to its otherwise great engine, *Turok* looked a lot clearer and cleaner. The textures were sharper and the enemies easier to see. To
attain such graphics with the N64’s limited ability to render a large number of polygons at once, the levels all had a steamy fog to them. Not only did this allow for the crisper environment, but it also gave the game a great sense of tension. A raptor could jump suddenly out of the fog and attack Turok.

The game also was a great deal more visceral. Enemies had many nasty death scenes. Some would grab their necks as blood spouted like a fountain while others fell to the ground coughing up their vital fluids. Turok had a very primeval feel to it, like running through a safari. Ambient sounds like birds, big cats, and crickets brought the environments to life. The features totally immersed the player into Turok’s fresh and new experience. To keep the immersion going, the controls became second nature after a bit of practice and were very accurate.

The next console FPS to take the world by storm again came from Rare. In 2000 they released Perfect Dark. Taking cues from their success with Goldeneye, Rare changed very little. In fact, in many ways its almost the same game. The two look and feel extremely similar. However, Perfect Dark is considered better for several different reasons. First, Perfect Dark utilized the N64’s new expansion pack. This pack added more memory in the system for the games to use. With this extra memory, Rare improved the texture quality and increased the AI. The catch 22 was that most of Perfect Dark wouldn’t run without the expansion. The single-player and half of the multiplayer environments just wouldn’t work. It was a pretty hard blow to many gamers, but the end results were very favorable. The game looked a lot cleaner and sharper, and the textures were more realistic that Rare’s previous offering.
One of the problems with *Goldeneye* is that it followed the film loosely and without any character development or storyline information. If people hadn’t seen the James Bond film, they would have a lot of trouble following what was going on. *Perfect Dark* was an entirely new story with more character development, story information told in cut-scenes. It used actual voices for the characters instead of plain text. The player still had to pay attention to follow the story, though. *Perfect Dark* also had a much better setting. In the distant future, aliens make contact with earth and allies with the agency for which Joanna Dark works. Another, more sinister alien race, allies with an equally evil earth corporation and the two sides come into inevitable conflict to stop the domination. The plot if full of some nice conspiracy storylines that, while not as in depth or exciting as say *Deus Ex*, are still really good for a FPS.

Where Rare shines again is also in the multiplayer. *Perfect Dark*’s multiplayer is setup almost exactly like *Goldeneye*, but with key differences. *Perfect Dark* offered a unique opportunity for those who grow tired of deathmatch all the time. Rare threw in a cooperative mode where two players could play the single player campaign together. It also added another element that proved very entertaining. One of the co-op options let one player play as Joanna and the other as various enemy soldiers. The player jumped from controlling one enemy to another in attempt to thwart Joanna from completing her objectives. *Goldeneye* only allowed four players to go head to head against each other. *Perfect Dark* replicated that with one major addition in the form of simulants. Simulants are players controlled by the game who’s AI can be adjusted by the player. The settings range from easy to perfect providing a wide array of choices. There are also simulants who can exhibit certain behavior. The TurtleSim is very slow but his shields are
extremely tough. The SuicideSim has no problems killing himself to terminate the player. There are many more variants available and add a great deal of fun and challenged to the multiplayer side of Perfect Dark. Although too many simulants on a map greatly lowered the game’s framerate (number of frames per second), they were the finishing touch to one of the greatest FPSs in console history.

Every so often a game comes around that shows influence from several of its contemporaries and while it does them justice, proceeds to show the industry something new. Volition’s Red Faction for the Playstation 2 is one of those games. Taking a cue from Half-Life, Red Faction is a highly story based game using the same plot progression style that eliminated levels and created a seamless undisturbed world. It shares many graphical similarities with Goldeneye and Perfect Dark with decent textures and interesting level design. While the animations weren’t nearly as smooth or life-like, the enemy AI was very intelligent for its time and for a console title. Troops would duck behind cover, run away to refill ammo clips, and regroup to take the player out. Running into a room with guns blazing was a sure way to get massacred. Strategy was a much better solution to most of the problems Red Faction tossed at the player.

The real draw for Red Faction was Volition’s Geo-Mod Technology. This Geo-Mod allowed the engine to create destructible environments not just randomly placed pots, glass windows, or crates but walls, floors, and ceilings. Can’t get through a locked door? Place a remote detonator on the wall beside it and blast your way in. There were several times throughout the game where after an intense firefight, the area looked like the aftermath of the lobby scene in The Matrix. Up until recently, the most damage any explosive would do to an environment was a black mark on the impact site. It was a neat
effect, but not even close to realistic. The level to which destructible environments heightens the fun factor of FPS gun battles immense. Unfortunately, Volition decided to implement destructible architecture less and less as the game went on and the cliché invincible objects that have plagued games since the beginning of the industry prevailed. Where Volition did use the Geo-Mod, it employed it very well and those sections went on to become some of the more memorable moments in gaming. One wonders why few developers have never tried to implement this kind of technology before, and many still ignore it to this day.

In 2001 Microsoft released its own console game system called the Xbox. Like many consoles before it, most of the launch games were lackluster and little more than demonstrations of the hardware’s capabilities. However, game developer Bungie had a trump card for this new system. Originally designed for PC and Mac, Bungie began exclusive development of Halo for the Xbox after being bought out by Microsoft. The result was nothing short of spectacular. Halo is considered one of the greatest FPSs since Half-Life and is definitely the best FPS on any console system.

Bungie was very ambitious with Halo on many levels. Where most FPS developers skimp on story to concentrate on action, Bungie spared no expense on either. Halo has one of the most compelling and engaging plots the industry has seen in years. The game is worth a run through simply for the well developed and voiced characters as well as the riveting twists. It starts in the distant future and earth has been at war for some time with a coalition of alien races known as the Covenant. The Covenant is out to completely obliterate the human race, whom they find an abomination to their god. In trying to escape, the last ship of an armada runs into a ring world called Halo. Before the
Covenant armada can overtake the ship, Master Chief a new cybernetic enhanced human warrior escapes with the ships AI construct, Cortana. To escape the Covenant and keep them from discovering Earth’s location, Master Chief and Cortana must unlock the secrets of Halo. Sounds simple, but it gets really intense and is superbly written and performed.

While the story is thoroughly entertaining, it’s the gameplay where Halo really shines. The Xbox controller has two analog joysticks, one controls movement and the other controls the player’s aim. While not as precise as a mouse, the setup is extremely smooth and fluid. It has a small learning curve, but once the player has them down they will never have to fight them again. In Halo, a player can only carry two guns at once. This dicey design decision could have easily backfired, but it turns out to be one of Halo’s strong points. If the player is running out of ammo for a particular gun, they can wait to find ammo and use their other gun, or pick one up from a fallen comrade or enemy. Since there is never a shortage of enemies, there is never a shortage of weaponry. This system creates some real inventive gameplay. Players need to know every weapon well and devise long term strategies. Few experiences are more exhilarating than having to jump from cover while reaching for a fully loaded weapon as Elite Guards are relentlessly attacking the player’s position. Bungie also gave the player a separate button for grenades, while every game up until this point forced the player to manually select them like any other gun in their arsenal. Grenades are support weapons, and in Halo they finally could be used the way they were meant to.

The Covenant forces were some of the smartest enemies in gaming history. It was almost like playing against living, breathing opponents. Hiding in one spot for too
long? Expect them to flush you out with a grenade or a charge attack. In close quarters where gunplay is harder, Elite Guards had no problems smacking Master Chief across his helmet with the butt of their gun (and you could smack them right back, making melee attacks in FPSs useful instead of pointless like in other FPSs). They will backpedal, sidestep, dive out of the way of explosives, out flank you, and jump from cover to cover to get closer to the player. The higher the difficulty setting of the game the harder it got, not only because of the higher damage counts, but because just when the player thought he had them figured out the Covenant taught him a new lesson in humility. As a result of this phenomenal AI, every time a player sat down to play Halo the experience was different.

Vehicles were something of an anomaly to FPSs, they were usually only used in third person games, where the player could see what he was doing. Bungie proved that vehicles could be used in FPSs and executed flawlessly. Running low on health or ammo and getting out numbered? Jump into a nearby Warthog (4x4) or a Covenant Ghost (hovercraft) and even the odds. Upon using a vehicle, the perspective shifted from first to third person giving the player supreme control and it was done with no loading or uncomfortable shift. It was just as easy as getting into a car and shutting the door. Not only was it a top notch addition to the single player, but Bungie also allowed players to use vehicles in the multiplayer. A Warthog has three seats with specific purposes, driver, passenger, and gunner. One person could drive while another fired out the passenger side and the last manned the machine gun mounted on the back. Some of the best multiplayer experiences with Halo had vehicles involved. So much so, many games released after Halo have included them in gameplay.
On the topic of multiplayer, *Halo* had one of the best experiences available. Four players could fight it out on one Xbox through split screen play like *Goldeneye* and *Perfect Dark*. However, players could connect up to four Xboxes together via network cables for 16 player deathmatches. Every Xbox needed its own TV and copy of *Halo*, but the reward was well worth the setup time. Deathmatches and teamplay were infinitely more engaging with the increase of players. *Halo* also offered a split screen co-op mode that allowed two players to play through the games single player.

All said and done, *Halo* was an unmatched console FPS experience and rivaled many PC FPSs not only in gameplay but also in graphics. *Halo* looked light years beyond the competition. If any complaints could be raised, it would be the fact that the level design of several sections of the game was somewhat less than creative. In other words, it was down right repetitive. Many locations looked exactly the same. On the whole, *Halo* can almost be crowned as the king of the console FPS. Its elements are already being implemented in countless games on all systems. Not since *Half-Life* has a game been so influential in the game development community.

Now there great FPSs on both console systems and PCs. With the rise of more powerful console systems many thought they would out distance the PC market and begin its decline. This hasn’t happened yet, nor does it even show signs of happening, but there is an ongoing discussion about which format is better for FPSs. Console Gamers never have to upgrade their systems to play new games that are released. The console controls are much better and more precise nowadays as *Halo* exhibits. As improved as the analog controls are on console controllers, they still cannot match the speed and precision of the mouse. *Halo* was recently released for PC and in almost every
review journalists commented that *Halo* was meant to be played with a mouse and keyboard. PC enthusiasts do have to upgrade their machines on a regular basis to keep up with all the high tech games that are released, but the resolutions are unmatched for top-notch system setups. The system, PC or console, that draws a gamer is a matter of personal preference. Gamers who grew up playing FPSs on the PC find most FPS console titles hard and inaccurate to control while console gamers find the mouse too sensitive. Hardcore gamers tend to adapt to either form of control, but many stick to what feels comfortable to them. There is no black and white answer, but the FPS started and evolved mostly on the PC giving the mouse and keyboard setup distinct advantages over analog control sticks.
Chapter 10  
The Clone Wars Part II  

Thanks to games like *Quake III, Unreal, and Half-Life* by 2000 gamers had experienced high powered graphics engines, visceral gameplay, and new complex storytelling. Just like the flood of games that attempted to emulate Doom’s popularity, the year 2000 began the second Clone War. FPSs became a dime a dozen when every developer tried their hand at creating one. Most were forgettable and many, while entertaining, simply copied traditional and cliché gameplay elements. Every other game seemed to be running off of the *Quake III* or *Unreal* engine creating a massive look-alike contest on game shelves. However several games did climb up the ranks by refining traditional formulas while experimenting with new ideas.

Raven is no stranger to the FPS genre with classics such as *Heretic, Hexen, and Hexen II*. In 2000 Raven gave gamers something that would spoil a good number of them forever. The best element of *Soldier of Fortune* wasn’t the hackneyed story with bad dialogue, nor was it the outdated modified *Quake II* engine that rendered it. It was a little thing called hit location. Enemies reacted realistically depending on where they were shot. Hit them in the hand, they would drop their gun. Shoot a leg, the character would jump up and down in pain or drag it along the floor. A head shot meant instantaneous death. Hit location had been used before, but not with the precision that Raven implemented in *Soldier of Fortune*. Plus, they didn’t just stop with realistic reactions; they loaded it with realistic damage modeling. That head shot had an entrance and exit wound complete with bone splintering inward and outward depending on where the bullet hit. Raven called it the GHOUL Engine and each model had several zones for depicting the damage. Blowing limbs off was as simple as using the right caliber weapon.
and precise aim, and as a reaction the blood splattered on the wall. Gamers had never been treated with this kind of realistic gameplay before and while the realistic damage modeling wasn’t a complete necessity to make a great game it added a nice visceral flair.

A few years later, Raven released *Soldier of Fortune II* and used the Quake III engine to keep up with the times. They also debuted the GHOUL II which improved on the hit location and damage modeling. The ‘gore zones’ as Raven called them, increased to over thirty and almost as many more realistic reactions to bullet hits. Now instead of half or whole limbs, players could blow off hands, ankles, and sections of the head. With the higher powered graphics engine they were rendered with even more detail. It’s a wonder why Raven never licensed the GHOUL engine out to other developers. Even today the *Soldier of Fortune* series is the only game series to have such realistic damage modeling for opponents. However nearly every game, especially FPSs use hit location for more realistic firefights.

It’s safe to say that the FPS genre takes itself very seriously. The games are all about action plots (with the few notable exceptions of *Thief* and *Deus Ex*) and fast brutal gameplay. Enter Monolith’s *No One Lives Forever* of 2000. In an era of intense FPSs, *No One Lives Forever* (*NOLF*) did something few games were able to; it had a very good sense of humor. It thoroughly spoofed the spy films of the 60s. You play as Cate Archer, an ex-Thief turned secret agent for the British agency known as U.N.I.T.Y. designed to combat international terrorism. A new evil organization known as H.A.R.M. (you’ll never find out what H.A.R.M. stands for, it’s a running joke through the entire game) rears its head and its up to Archer to discover their plans and thwart them. It’s the most humorous, laugh-out-loud story gamers had seen in a long time. It never took itself
too seriously, created some truly memorable characters, and delivered the dialogue very well. Who wouldn’t find humor in the constant bickering between a drunken Scottish henchman and an overweight Swedish Opera singer that can’t carry a tune? Guards will also talk to each other about the merits of working for an evil organization and exploring philosophical topics like why people turn bad and why more women aren’t involved in crime. Some become so likable that the player actually feels guilty about killing them. They feel like real people stuck working in an evil organization because it was the only job they could get.

The humor and style doesn’t end there, NOLF has some of the coolest gadgets in a game. Archer has a barrette to pick locks, sunglasses that can see infrared, sleeping gas in perfume bottles, and lipstick bombs. For the sequel, No One Lives Forever 2: A Spy in H.A.R.M.’s Way Monolith followed the same formula of humor and lighthearted story as well as improving the gameplay on nearly every level. Its so hard not to laugh when setting a kitty proximity mine (which appears to be a kitten) and hearing it meow until a female ninja gets too close and then it hisses and explodes. This series is overflowing with a wit and style that is practically unmatched in the FPS genre. Fighting a group of French Mimes with tommy guns is a refreshing change of pace from every other hyper-realistic FPS on the market.

A game doesn’t necessarily need a state of the art graphics engine to produce an engaging and new experience. Clive Barker’s Undying is one of those games that uses an outdated engine (Unreal) but still manages to pull itself ahead of all the other offerings on the shelf. Undying’s strength lies in its atmosphere and superbly told tale of horror. Few games deal with the horror setting and even fewer actually succeed in creating a
truly scary experience. Electronic Arts (EA) utilized the *Unreal* engine to its fullest and created a world overflowing with style and detail. Within this world Irishman Patrick Galloway returns to his friend Jeremiah Covenant after receiving a letter from his comrade. Apparently ever since returning from World War I, the Covenant family has died off one by one. Jeremiah’s brothers and sisters have returned from the grave to complete a curse they all were a part of since childhood. Jeremiah is the last one left and too weak to investigate and stop the curse. Galloway agrees to help since his friend had saved his life in the war. It plays out exactly like a horror novel, and since it was written by Clive Barker its filled with complex twists and disturbing turns. Truly a masterpiece of horror, Undying never lets up from beginning to end.

*Undying* also employed an interesting weapon system. Galloway could fire weapons from his left hand while casting spells with his right. This system proved to be very useful and entertaining. The player no longer only had guns to fall back on. If being overrun by Howlers, the player could cast the spell Shield to defend against the attacks while switching to silver bullets for the pistol to take them out. Offense could be mounted at the same time as defense instead of having to switch between them as in other games. This unique two weapon system proved right at home in Barker’s 1920s horror themed game.

In 2001 a Croatian game developer called Croteam unleashed an FPS that would give gamers sore fingers and wrists for months. While not as satirical as *NOLF*, *Serious Sam* definitely didn’t take itself seriously. It made no mistake at what it was trying to do; entertain the player while giving them a workout. *Serious Sam* took gamers back to a time when gameplay involved nothing more than running though a level trying to find the
exit and mowing down anything that got in the way. But *Serious Sam* did this with a twist. The Serious engine used to make the game was not only one of the best looking graphics engine out at the time, it could also render literally hundreds of enemies on the screen at once. This game was not for gamers who dislike a constant challenge. *Serious Sam* was full of epic battles rivaling the Battle of Helm’s Deep in *Two Towers*. Just when the player thought it was over, Croteam had more waiting. *Serious Sam* was a test of endurance and reflexes. No other FPS has ever thrown as many monsters at the player at once only to do it all over again the next second.

Many games advance plot through cutscenes rendered using the game’s engine. During these advancements the player sits and watches the action unfold without interacting. 2015’s game *Medal of Honor: Allied Assault* did things differently. The player was never sitting static simply watching the action; they were always a part of the action around them. In the opening level an Allied convoy is trying to sneak into a German base to rescue a captured comrade. The player is in a jeep with other soldiers waiting while an officer checks their papers. The driver gets too nervous when the soldier argues about the paper’s validity and shoots the officer. All during this time the player is free to look around the van and out the back. Now since the soldier’s cover is blown everyone jumps out of the van and the player can jump out with them and storm the base. All of *Allied Assault*’s sequences use similar openings to quickly immerse the gamer in action. The face of the player character is never shown in any scene. The person playing is the soldier running through a Europe entrenched in World War II.

Many of *Allied Assault*’s action sequences are scripted, that is the events that take place only have one possible outcome. No matter how many times the player replays a
scripted section, he cannot change what happens. While this may seem like cheating the gamer out with linear gameplay, these sequences are extremely well composed. They feel less like linear play and more like involving the player in the middle of chaos that he cannot always control. The gamer is a part of the action, but not the driving force. This technique really brings the game to life and puts war in a different perspective. *Allied Assault* with always be remembered for its recreation of the Normandy Invasion at Omaha. It starts with the player immersed among fellow troops being shuttled to the shore. While looking around, one can see other soldiers puking, clutching their weapons in prayer, and mortar shells pounding the ocean around them. Upon reaching the beach the hatch opens and nearly every soldier in front of the player is cut down by machine gun fire. Running up the beach to the bunkers is absolute chaos as bullets whiz by while the gamer ducks and runs from barricade to barricade. All around him soldiers are dying, medics are screaming, and shells are exploding. It is the single most thrilling and startling level in the entire game and other games of the period. It is like a scene in *Saving Private Ryan* without the gore (taken out to reach a larger audience).

A few years later, members of 2015 left and created the company Infinity Ward. In 2003 they helped gamers revisit WWII again with *Call of Duty*. *Call of Duty* was a lot more action packed and had less scripted events than *Allied Assault* yet it immersed the player to a greater degree in WWII battles. Only three weapons could be carried at once to heighten the realism. However any enemy and ally weapon could be picked up from the ground and used. *Call of Duty* also covered three nation’s campaigns (American, British, and Russian). Infinity Ward treated the Battle of Stalingrad with the same prestige as Omaha. It begins with the player on a boat with his comrades, without guns,
ammunition, and officers threatening that traitors will be shot. After a few dive bombs by planes a few Russian soldiers jump ship and are consequently shot by their superiors. Upon landing the player is shuffled into a line, handed a pack of bullets, told to use the rifle of a fallen comrade, and thrust into German machine gun fire. The player will spend this entire level without a gun. This section was definitely inspired by the film *Enemy at the Gates* and reproduced with thrilling unmatched results. The battles from *Allied Assault* and *Call of Duty* will be remembered for years to come.

In 2002 Epic returned to the FPS gaming scene with their next installment of *Unreal Tournament*. *Unreal Tournament 2003* was a lot like the original in premise and gameplay only sporting the new graphics engine created by Epic. The engine was more gorgeous than any other engine on the market at the time, but it wasn’t only the engine that wowed enthusiasts. *Unreal Tournament 2003* sported a new entry into physics representation called rag doll. Rag doll physics get rid of scripted animated death sequences relying on computations that depend on where the target was hit and with what magnitude of force. If a target was shot in the head it would slump to the ground. If the target was shot on a hill, it would slide down bouncing off rocks and other obstructions. A shot to the shoulder would send an opponent spinning backwards bullet wound first. Rag doll deaths add much realism to combat, unless overdone. Some games use the physics capabilities too extensively allowing limbs to bend to degrees not possible in reality. It is a first step to utilizing believable physics. Nowadays a growing number of games are using Rag Doll physics because of the more realistic depiction of shooting deaths. As physics engines get more advanced, game worlds will become increasingly engaging and representational.
What Does the Future Hold?

December 10th 2003 marked an anniversary in the gaming industry. id Software’s *Doom* turned ten making the FPS genre a little over a decade old. The question has been raised by industry journalists of whether or not gamers have seen everything the genre has to offer. Shelves have been flooded for years by mediocre FPS games and even some of the traditionally more innovative titles are only recycling gameplay that has been used for years. Where are the innovations and experimentation? Does this genre have any promise of innovation or will it be doomed to revisit outdated formulas? Several industry developers disagree and say that there remain many exciting hurdles to tackle as technology gets more powerful.

Warren Spector, creator of *Thief, System Shock,* and *Deus Ex,* says that content and interaction is where the future of gaming is. Graphical prowess is already being pushed to the limit of what developers can do. The real challenge and future of FPS gaming, according to Spector, is in the player’s interaction with the game world and its inhabitants:

“I hope games...point toward a future in which interaction between player and game world becomes much less constrained than it has been in the past. Solving puzzles can be fun, but making your own plan, finding your own solution, having your own unique experience is more satisfying, I think. More detailed worlds HAVE to allow more varied interaction -- if developers just use processing power to create prettier backdrops, we're in trouble (Gamespy, “Warren Spector on the Future of Gaming”).”
Spector suggests that this extra processing power should be dedicated to portraying convincing and realistic physics, AI, and sound. He points out that AI should not be synonymous with combat behavior. More time should be spent creating a broader range of human behaviors for non-player characters. Physics should also be given special attention because from a gameplay standpoint, highly realistic physics simulation allows for a wide variety of problem solving possibilities never before available. Sound is also one of the aspects of FPS games that often doesn’t get as much attention as it merits. With increases in sound card technology and an increasing number of people with positional sound systems connected to their PCs, more attention needs to be given to the essential element sound in games. “Positional sound cues will become an ever more important part of the player feedback system -- alerting players to what’s going on in the world and what impact they’re having on it...”

Game developers today are attempting to exceed their prior goal of simply eye-popping graphical engines. Valve Software is hard at work on their next title *Half-Life 2*, which will incorporate many of the points Spector mentioned. They have developed what looked to be almost the most advanced physics engine ever seen in a game. Every object in the game has simulated mass and weight and reacts realistically when manipulated. In a demo at the Electronic Entertainment Expo in Los Angeles Valve’s Doug Lombardi showed off many of the game’s advanced features. Using an anti-gravity weapon, Lombardi picked up a body by the leg and moved it around a room in the game. The body simulated weight and movement like it would in reality (rag doll) as it bounced off of surfaces and knocked a table over which spilled all of its contents on the floor. He also explained that their engine can tell the difference between objects composed of
different materials. Bullets and objects would react differently when they came in contact with wood, metal, tile, or brick and demonstrated this by splintering wood with a crowbar and ricocheted bullets off of tile and metal barrels each with a unique sound depending on its composition.

Valve’s Managing Director Gabe Newell demonstrated the highly evolved AI that the player will work with and against in *Half-Life 2*. No event in the game is scripted like in *Allied Assault*. Instead, ally and enemy AI will look at the situation and decide what the best course of action is. This was demonstrated as Gordan Freeman, the game’s protagonist, was being chased by troops. Upon running into a room, Freeman closes the door and blocks it with a table. A troop then tries to get through the door by pounding on it, when that doesn’t succeed the troop breaks a nearby window and sticks his gun in and blindly sprays the room with bullets. Events like these will be commonplace as the AI traverses the environment and reacts to challenges they come across.

The father of the FPS genre is also stepping up to take gaming to the next level. A little over a year ago id announced they were working on *Doom 3*. The game is set to retell the story of the original *Doom* moving from purely action to a cinematic style of gameplay focusing on the single player experience. One glance at a screenshot will reveal what all the excitement is about. Lead programmer John Carmack has created the most amazing graphics engine ever to grace a game. Changing from polygons to bump-mapped textures (applied to a model to give it a smooth 3D textured look like a relief map), the game’s characters and environments are absolutely unmatched. But as discussed before, graphics are a plus not a necessity to a groundbreaking game. It seems id is well aware of this and they are not only pushing the graphical limits, but the ideas of
environmental interaction as well. *Doom 3* is sporting the same advance set of physics simulation as *Half-Life 2* (though using a different engine), but uses it differently. Where *Half-Life 2* is action based, *Doom 3* is set to terrify the player like never before.

Along with the incredible physics, *Doom 3* will support dynamic lighting, so light will be rendered in real time within the engine. Knock over a lamp and the light will dynamically change as it falls to the ground. If a ceiling light is bumped, the swinging lamp will send light in all directions and create realistic shadows from obstructing objects or creatures. Until the demonstration at the Electronic Entertainment Expo of this year, effects like these have never been seen. Muzzle flashes will also treated as light sources. Imagine walking into a room on the Mars Base and suddenly the lights go out. Hearing a noise behind you, you spin around and fire. The flash of light from the gun reveals a zombie and with each gun shot it lumbers closer and closer. Light is going to be used in this fashion throughout *Doom 3* and the demons will use the shadows to their advantage. They will also be able to crawl on walls and ceilings as well as break through doors and walls to get to the player. The days of waiting behind a door for a following enemy to appear are gone. These relentless enemies might decide to break through the wall instead. Unlike *Doom*, *Doom 3*'s engine is not designed to throw countless demons at the player. The focus is to create the most horrific atmosphere where ammunition is rare, demons are tough and huge, and the player is all alone. Based on the demonstrations from id, they are well on their way to reaching that goal.

For FPS gaming, the future looks bright and promising. Developers like Warren Spector, Valve, and id are interested in taking the challenge to advance the experience to the next level. They are painstakingly trying to utilize every piece of hardware available
to create the most engaging games on the market and provide players with immersive moments they’ve only dreamed of. While it’s hard to find new ways to express old formulas that have been around for the better part of a decade, there is still much possibility to be realized. Graphics can increase exponentially, but AI, physics, sound, plotlines, and gameplay can be integrated into a game to create experiences that will be unique to the person playing. One thing is certain, FPSs are here to stay. The FPS remains one of the most popular perspectives in gaming and developers are working very hard to keep up with the competition and to fulfill the increasingly intense demands of gamers.
Bibliography

Books


Game Review and News Websites

http://www.3dactionplanet.com

http://www.acornarcade.com

http://www.gamesdomain.com

http://www.gamefaqs.com

http://www.game-revolution.com

http://www.gamespot.com

http://www.gamespy.com

http://www.ign.com

http://www.swgamer.com

Games


Counterstrike. Sierra Entertainment, PC. November 2000.


Doom. id Software, PC. December 1993.

Doom II: Hell on Earth. id Software, PC. September 1994.

Doom III demonstration. id Software, PC. May 2003.


Half-Life. Sierra Entertainment, PC. October 1998


Quake. id Software, PC. May 1996.

Quake II. id Software, PC. November 1997

Quake III Arena. id software, PC. November 1999.


Unreal. GT Interactive, PC. April 1998.

Unreal Tournament. GT Interactive, PC. November 1999.
