

Michael Chang
STS 145
Case History
March 22, 2001

H A L F – L I F E *Everywhere!*

One cool, fateful day in March of 2000, a college student unleashed a video game on an unwitting and innocent dorm's all male floor. Bright students, virtuoso musicians, champion athletes, and master gamers alike filled this first floor of guys with not only an odor characteristic of junior high and high school locker rooms but also a harmonious marriage of profanity, grunts, screams and gunshots. The profanity, grunts, screams were courtesy of the guys on the floor, and the gunshots courtesy of that video game. The story begins when on the third floor, the tamer co-ed floor, two sophomores had found pirated copies of the game, installed it, played it through to the end, and began dabbling in its multi-player mode. They fought each other day in and day out until their games felt empty, devoid of a sense of community. They wanted more targets, and they wanted more fun, so they made the game available to residents on the first floor. As soon as a player configured his keyboard and mouse to his liking, and as soon as that player dismembered his first opponent with a gun, the game was on. In no time the game attracted half of the residents on the floor and even spawned an intra-dorm contest among players. Just as quickly, the game's title substituted for the actual act of playing the game: "Wanna Half-Life?"

Why "Half-Life" in particular? For all the other similar games out there, we always came back to this game because of its accessibility and the appeal of interaction. More or less, "Half-Life" fulfilled our recreational wants. Comparing the lure of "Half-Life" with that of other activities, we discovered we could accomplish most, if not all, goals of those games and

activities in the simple act of convening everybody in the same medium at the same time. In short, this multi-player game, with its speed, action and free and casual discourse, condensed all the recreational elements of other activities into the game's structure. Although others opted to play sports for their fills of fun, such was nearly impossible for the student burning the midnight oil or for the group of slothful kids in the lobby who considered the lounge ten feet away too far to go. Especially, through the first-person perspective, a player could vicariously experience physical exertion as if in war or a game of paintball. Among other traditional games, only multi-player board games, around which over 10 people can sit and compete, offered similar reach and enjoyment. Even then, with the compounded testosterone on the floor and stress and frustration accumulating through weeks and weeks of study, board games offered little in the way of channeling one's emotions or relaxing the mind. Mindless violence and gore was the way to go. Beyond that, players welcomed the escape into a virtual world to explore beautiful maps and innovative technologies at play. To top it off, nothing short of sleep offered more comfort or proximity than a video game played in one's own room.

In late 1998, "Half-Life" emerged as the latest success story from a slew of first-person shooters (abbreviated as FPS) such as id Software's Doom or Quake series. "Half-Life" can boast of being one of the most lauded games of all time, with over 50 "Game of the Year" and similar awards (SierraStudios.com). No less, it has inspired such extreme comments as "this game restored my faith in gaming" (Gamespot.com via SierraStudios.com). At face value, "Half-Life" just seems like another FPS, a repackaged version of Quake. However, familiarity with FPSs alone does not suffice to attract gamers to a new game. How a new game in a worn genre succeeds depends on its ability to seamlessly allow many players to participate all at once and to surprise its audience, in particular the jaded and cynical computer scientists that

comprise the “hard core” gamers. Designed and developed by Valve Software and published and distributed by Sierra Studios, “Half-Life” packages two significant enhancements to the FPS genre, an innovative single player action game and an accessible and flexible multi-player game. Although “Half-Life” drew originally for its single player game, it owes its enduring presence to the multiplayer.

Inspired by id Software’s triumphs and licensing Quake’s original code, Valve released “Half-Life” as its entry into gaming. Founded in August 1996 by Gabe Newell and Mike Harrington, former Microsoft engineers, the company decided to first release “a massively multiplayer first person strategy combat game,” according to Ken Birdwell (Interview w/ Ken Birdwell, March 8), Valve’s Senior Software Development Engineer and one of the lead “Half-Life” designers. However, reluctance to delay its entry to market and a shortage of technical knowledge pressured Valve to build a more derivative game. Harrington’s extensive network of contacts included colleagues at id Software, so he assured Valve a license to the original Quake technology (Interview-Birdwell). In turn, the development team began their own FPS with the hopes of not “writing a lot of code and not taking too long” (Interview-Birdwell).

“Half-Life” was born from the idea of throwing more monsters and bigger levels at the first person than Quake or Doom had. Radioactive experiments and clandestine government supervision all pointed to one thing: encounters with a different kind of life form, certainly not an original concept. However, the lifted code and the lifted plot arc accelerated the development process, and less than a year after Valve’s plunge into the FPS genre, it had a releasable prototype for “Half-Life.” At the time, the multiplayer was only a forethought, and according to Software Development Engineer and one of the lead designers of the multiplayer code, Yahn Bernier, “the focus when we shipped Half-Life was 99% on the single player

game” (Interview w/ Yahn Bernier, March 16). Nonetheless, “Half-Life” in any shape or form was to be the next in a field of aspiring “Quake Killers” (Interview-Birdwell) such as Daikatana, set in the distant future featuring time travel and magic, and Sin, a biotechnologically themed game with a blue chip company as an enemy. Unfortunately, with its upstart roots and less than coincidental parallels with the original Quake, Valve confronted bad press and negative buzz among the influential gaming industry. Common jabs at the team included: “Half-Life: I guess they couldn’t afford a full one!”(Interview-Birdwell) Indeed, following requisite self-critiques of their design, Valve found the game a Quake clone at best and judged it unable to survive in the crowded market for FPSs (Birdwell article).

As most of Valve’s engineers also wore the owner’s or the executive’s hat, they decided to extend their deadline at least a few months to tweak the game and in essence, make it more “fun”(Birdwell). Although creating “Half-Life” from scratch took only a year, the overhaul that would follow took another full year in itself. During this critical phase, Valve discovered ways not only to improve the game from a technical standpoint but also to mold the game to the developers’ tastes. Avid gamers themselves, if they found the game fun, surely others would as well. Soon they found flaws not only in the game but also in their approach to designing it. Their original approach could be described, at best, as haphazard. A disjointed team with ideas shooting off in different directions, Valve got itself nowhere that any other gaming company could not get itself either.

Therefore, higher level engineers decided to pool their resources and produced the “Cabal.” A group of computer guys more than anything else, the Cabal “combined the strengths of a cross section of the company”(Birdwell). Element by element, the Cabal re-evaluated the entire game on their hands, and “once enough ideas were generated, they would

be reorganized into a rough storyline and chronology”(Birdwell). To ensure uniformity in the game and in the designers’ minds, the Cabal convened often: four days a week and six hours per day. As they strung together ideas and the overlaying structure and purpose of the game, its re-development fell together quickly. Within three months, by the time play testing began and smaller tasks needed to be assigned, the Cabal gradually receded from the design process, it’s work having been accomplished. In fact, contrary to their initial apprehension about the group dynamic, “the people involved were tired of working in isolation and were energized by the collaborative process, and the resulting designs had a consistent level of polish and depth that hadn’t been seen before” (Birdwell).

In support of his company, Newell championed the “fun” campaign with his observation: “You can’t show the player a really big bomb and not let him blow it up” (SierraStudios.com) Sure enough, Valve noticed that “Half-Life,” as it was, restricted players from interacting with the engine the way they wanted. Shooting some parts of monsters but not others would cause harm, making one’s fate seem arbitrarily set by an enemy’s boundaries. Even worse, players lacked total interaction with their environment. A random portal to another dimension was easily accessible, but a wooden crate was immune to any variation of detonations, blasts and contact. The multiplayer did not even exist yet. By extension, players would have blamed the company for these shortcomings, and as word of mouth spreads, so too would the game’s demise. As a result, Valve reprogrammed the game to fulfill players’ needs. In short, the game responded to a player’s actions and to his expectations. Here, an interesting distinction must be drawn. Although these details fulfilled a player’s expectations, the game as a whole still managed not to pander to the player. If it had, the game would be a predictable rehashing of players’ gaming experiences rather than a new challenge surpassing their

imaginations. Indeed, “Half-Life” is rife with plot twists and surprises. For one thing, at its midpoint, the government returns to the scene of the disaster only to erase these experiments and discoveries in the face of public exposure. The protagonist, Gordon Freeman, finds his human friends now foes.

Designers set two goals for themselves: to make a fun game and to make a lot of money. (Interview-Birdwell) One goal could not be accomplished without also accomplishing the other. If they made a fun game, they would also make a lot of money. At the same time, they would not know if their game was fun to anybody unless they made a lot of money from it. Their development of the game guaranteed nothing because neither money nor “fun” can be engineered. However, the undying business philosophy within the eclectic team was that a good game sells itself. Birdwell attributes 90 % of a game’s success, i.e. its sales, to word of mouth, and inversely, a game generates word of mouth only through quality. They strove to make a “top 10” game and in so doing, a lasting impression on the industry. The justification was this: “Given the amount of time and effort a bad would take, none of the core people at Valve wanted to waste their time on anything other being a top 10”(Interview-Birdwell).

As designers and as managers, the Valve team afforded itself the chance to do business the way it thought it should be done, with the intent of meeting expectations. From their unprecedented success, there is a lesson to be learned from Valve’s business model. Ultimately, with its heart in customer satisfaction, Valve foretold its own success. Since its release, “Half-Life” has established an excellent precedent to help game developers who are used to sacrificing the quality of their games for a quicker release. As Birdwell notes, “There are a lot of talented people who could have made great games, but are crushed by silly decisions in upper management. If we’ve given people a way to fight that, then I think we’ve

done a good thing.” The history of gaming has seen hundreds of carts for the original Nintendo Entertainment System (NES) become blockbusters simply by virtue of Nintendo’s publicity channels like Nintendo Power and a stable of Saturday morning cartoons. The majority of those games, however, lacked substance and replay value. As a startup software business with little history and limited access to high caliber marketing, Valve had no such fortune. Therefore, as “Half-Life’s” original build teetered on the edge of falling toward those games, Valve was not about to let the game release for the sake of rushing it to market.

Consequently, they derived an efficient response system that allowed for constant modifications until the eve of its release and key feedback from a representative audience that specifically included young-to-middle-aged males (Interview-Birdwell). To that core demographic, the game had to strike a chord, and in the end it was that demographic’s input Valve used for re-tooling “Half-Life.” Incorporating “play testing” into their self reviews of the game, Valve answered future customers’ concerns by affording them the very opportunity to test the game before its release. Moreover, because they felt that the group of pre-release testers represented a fair cross section of their potential audience, the designers stuck to the same group for most of the 200 sessions of testing. Another step in the convergence between different entertainment media, this testing closely resembles movie studios’ efforts to preview movies and gauge audience reactions to them. Via response forms and comments, movie makers tweak their film the best way they can, not at all different from Valve’s approach.

The results of these concerted efforts are immediately conspicuous and therefore, all the more appreciated. As a new standard for PC FPSs, “Half-Life” wed ease of use to technological leaps. Accessible by any computer and any player, one of “Half-Life’s” finest attributes is its egalitarian position. For all its innovation, the game managed to be inclusive,

allowing for backward compatibility. Careful not to alienate potential players, the developers included software rendering in the game so that older computers lacking sufficient memory, resolution capabilities or 3-D processing could still support the game (Planet Half-Life). Therefore, even players without the bucks for computing power retained the spirit of “Half-Life.” Although this feature did little to balance disparity among computers, it did at the same time allow any player to try the single player or to join a large network game and still compete. Even players on the fly could connect their laptops and join games.

To seize on new technology, however, “Half-Life” reached beyond the bounds of software code to tap into hardware advancements. Aurally, Valve integrated full digital signal processing (DSP) capabilities and multi-channel sound representation into the game’s technology. DSP could distort a prescribed set of sounds to accommodate any environment, so that the same basic sound played in different areas would sound slightly different (Planet Half-Life). To this end, the programmers allowed themselves to keep the total number of sounds and sound files to a minimum and instead, command the DSP technology to alter it when appropriate. In addition, consistent sounds allowed players to learn quickly and efficiently (Planet Half-Life). Finally, with realistic sound and accurate mapping of sounds to their ongoing actions, a player could further immerse himself within the gaming environment.

Graphically, the game exhausted Intel’s multimedia extension (MMX) technology for PC microprocessors and 3dfx’s new graphics board standards (Planet Half-Life). At once, any casual gamer will notice the game’s fluid animation as a vast improvement on predecessors’ graphics engines. The single largest factor was Birdwell’s revolutionary Skeletal Animation System. In essence, the system maps each moving object in the game not as part of the whole frame as in Quake but as its own discrete object. The animation system polishes those objects

with finer detail and better defines them against the sprawling backgrounds. Moreover, drawing discrete objects removes restrictions of fixed frame rates from the animation process, allowing for smoother rendering of moving objects (Planet Half-Life). Monsters appeared smoother and larger than ever before, and the gore authentic and plentiful.

Beneficiaries of the technology included all million and a half gamers with copies of “Half-Life” in addition to the game’s countless pirates. Although myriad enemies and levels appeared brilliantly for the single player, the technology translated even better in the multi-player mode, where maps and all their crucial objects served as the only source of variation for players. Moveable obstacles taking advantage of the skeletal animation system figure prominently into a player’s strategy. For example, one player could move several crates to partially block an entrance, and once another player distracts himself trying to navigate around them, the first player can easily dispose of the second. More important, the skeletal animation system also applies to character animation, so that players are more visible and obvious to each other, reducing the chameleon effect which unfairly allows a player’s skin to blend into the background. Finally, the smoother animation at least partially offsets a computer’s lag, so that games can still be rendered effectively over large networks.

“Half-Life” also debuted a network interface beyond any other. What stood out about it was not its improvement on previous systems but its complete overhaul of the multi-player FPS. To prevent any kind of miscommunication between a cold gaming interface and the fickle user, Valve completely streamlined the process of starting a game, which usually can be the most difficult barrier to people playing it. After all, what kind of American gamer with a short attention span is going to play, continue or join a game if he has to wait several minutes or navigate innumerable interfaces? To start, the Valve team decided to strip down the extra

windows and integrate multi-playing into the main client (Interview-Bernier). Therefore, whenever a gamer would start up any of a number of “Half-Life” games, he could merely open the main client. From there, he could resume a single-player game or join a multi-player game. Choosing the latter opened another tree of options: a Local Area Network (LAN) game, an Internet game or a host of configurations to edit one’s multi-player profile.

To join an online game, a gamer only had to double click on the server he wanted to join. Rather than enter an Internet Protocol (IP) address to find an individual computer or run a separate program to access outside networks, “Half-Life” bypassed those steps and offered immediate satisfaction for the gamer. One step farther, the multi-player game even featured a “Quick Start” button that directed a gamer instantly to a responding server. The only other company with “anything similar at the time was GameSpy, but we weren’t satisfied that they could better serve our customers than we could in our own client” (Interview-Bernier).

However, like all violent games, “Half-Life” had to confront its place in society. Addressing the heated issue of violence in media, Birdwell noted that Valve did not even consider the issue of violence in making the game, simply assuming the gore. In fact, although a “Content Control” filter is built into the game, it exists not for the American public’s concern about violence but to circumvent certain countries’ restrictions on imported media (Interview-Bernier). The case against excessive gore in such games as “Half-Life” states that it is a bad influence on the minds of impressionable people. That gore, in turn, might encourage them to try to reproduce similar effects in real life. However, under close scrutiny, “Half-Life’s” violence can rarely be confused with realism because it is so outlandish. Only in the cases of shotgun blasts or magnum shots does the game begin to graze reality: a loud shot is fired, and a splash of blood flies from the victim. On the other hand, some situations can betray reality just

as much as those moments resemble it. With the gauss gun, one can propel oneself across an entire room or shoot through any thickness of wall. Such a weapon and its rampant use typifies fantasy violence at its best, and no doubt, any gamer knows this all too well by now. By the time “Half-Life” released, “there were other more obviously gratuitously violent games available so our PG-13 action-move level of violence makes an odd, and easily defensible, choice to be attacked”(Interview-Birdwell).

Actually, the game’s excessive gore can be seen as a preventative measure. The violent aspect of “Half-Life” serves to absorb the frustration that could actually lead to real violence. Like an additional backdrop or weapon, the gore merely adorns the game, suggesting violence might only be an inconsequential part of the game. Nevertheless, gore not only provides confirmation of the kill and validation of one’s ability but also contributes to the overall satisfaction of gaming. With such blood, guts, spillage and dismemberments, the kill is unmistakable. Such explicit kills, affectionately labeled as “frags,” also reinforce both the player’s willingness to continue play and his self-confidence. In the end, surpassing another player in the multiplayer with a devastating kill satisfies like no other game, and so, the ends do justify the bloody means.

“Half-Life” now finds itself an integral part of people’s daily routines. Such a schedule is no less normal or no less hectic than a schedule with a sports practice, music practice, drama rehearsals or club meetings. Thanks to its flourishing multi-player, people around the world can enjoy the flourishing community of which they are a part. To partake in a game with thousands of others daily at any hour or energy level of the day only heightens that sense of belonging to the reclusive gamer. For the casual gamer, “Half-Life” represents the ultimate escape because of its simple-minded goal of killing the opponent. Even Dennis Fong, the

epitome of competitors and a champion at multiple FPSs, offered in lecture that he has “good friends” from online gaming communities and keeps in touch with them on a regular basis (Dennis Fong lecture). No doubt, the game has had a profound impact on social gamers if not because it is a revolutionary game then certainly because it introduced a more competitive and interactive game.

Since its release, “Half-Life’s” creators have completely shifted their focus to the multiplayer game, with new priority on a “robust mod-making” tool based on the “Half-Life” engine (Interview-Bernier). Valve and Sierra employ the “Half-Life” engine to build still more spectacular FPSs like *Gunman Chronicles* and *Team Fortress 2*. However, as new “Half-Life” games keep surfacing, the original game, its multi-player follow-ups like “Counterstrike,” “Team Fortress,” and “Opposing Force”, and their base of players continue to proliferate. In total, gamers spend 1.7 billion minutes a month on “Half-Life,” and at any given time, one can find 65,000 players online (Interview-Bernier). Even now, a visit to Sierra’s website yields several versions of the original game and its siblings still for sale.

No doubt, “Half-Life” has elevated the potential of game design not only in its attention to detail but also in the way it assures its audience of having fun. On so many levels, the game works. It is an involving single player game, a limitless multi-player game, and an easily rewrite-able engine to accommodate the varying tastes of global gamers. To this end, “Half-Life” has impacted the way games are designed, the way business is done and the culture not just of gaming but also of the entire cross-section of society that engages in interactive games. Take a stroll down an all-guy floor on a college campus and bask in the screams and yells of the computer geek and the starting quarterback alike as they hunch over a computer screen and point-and-click their way to a gory, 3-D explosive bliss.

WORKS CITED

Bernier, Yahn. "RE: research on Half-Life." E-mail interview to Mr. Bernier. 16 Mar. 2001.

Birdwell, Ken. "The Cabal: Valve's Design Process For Creating *Half-Life*." Gamasutra – The Art and Science of Making Games 10 Dec (1999). 10 Mar. 2001
< http://www.gamasutra.com/features/19991210/birdwell_pfv.htm >

Birdwell, Ken. "RE: some more questions about Half-Life." E-mail interview to Mr. Birdwell. 8 Mar. 2001.

Fong, Dennis. Lecture on competitive gaming and experience in First Person Shooters. Stanford University, Stanford. Feb. 15, 2001.

Gamespy. Planet Half-Life: Half-Life Information. 11 Mar. 2001
< <http://www.planethalflife.com/half-life/guide/overview.shtm> >

Sierra Studios. The Official Half-Life Web Site. 4 Mar. 2001
< <http://www.sierrastudios.com/games/half-life/> >

Interview with Yahn Bernier

First of all, was anything in the single-player game or its story sacrificed to accommodate a multi-player engine?

No, the focus when we shipped Half-Life was 99% on the single player game. The original multiplayer component was a showcase for the HL weapons, but as few changes as possible were made. The engine all through development was kept strictly client / server, making reworking the underlying networking pretty transparent to the game code, so that turned out to be not too terribly hard to do. There were no changes to the story for HL based on the multiplayer component.

Was there anything in the multi-player mode added or subtracted for business reasons, eg. if the weapons were too unbalanced, if the map-designing was too difficult, etc?

Yes, several of the weapons were tweaked to make them more balanced for multiplayer (the Gauss gun kickback for instance). The multiplayer maps were created solely for the multiplayer mode.

How was the importance of a good multi-player game weighted against having a good single-player game?

The priority was on single player through shipping HL. Then the priority shifting to providing a robust mod-making SDK and the test case for that was Team Fortress Classic. TFC was critical to establishing the HL engine as a great mod-making platform.

Which game was more stressed by the upper management of the company?

Through original shipping of HL, the single player, from that point forward, 100% on the multiplayer game.

On the technical side, how did "Half-Life's" multi-player networking technology work in making it so much easier for gamers to join an Internet or LAN game than before?

We took a look at the current state of the art, which requires you to manually type in IP addresses or use external applications (with very complicated user interfaces) to connect to multiplayer games. We knew early on that this was a really bad model and that it would be better to a) simplify the connection process and b) integrate all browsing into our client directly. To this end, we shipped with a "Quick Start" single button "find me a game and connect me" button and we shipped with an integrated

server browser. To my knowledge, we were the first on-line action game to do this. Later we realized that it was important to support browsing for mods in the client, so we added the "Custom Games" menuing system to manage and download user mods from the net.

Was this technology completely new and invented by your team, or was it built from licensed technology somewhere else?

All of the client stuff was completely new (to us at least). The only company doing anything similar at the time was GameSpy, but we weren't satisfied that they could better server our customers than we could in our own client.

With such success for the multi-playing interface for the death-matching game, Counterstrike, Team Fortress, etc, will Valve and other companies be sticking to something similar in coming years and upcoming products, or are you continuously tweaking and improving it?

A little of both, we've learned a lot and are always rolling those changes back into the core platform features.

As for the public's concern about violence, I understand there's a gore filter built into the game. When was it decided to have that built in, and do you think that affects the success of the game or the fun of death-matching?

Actually the Content Control feature had nothing to do with the public's concern about violence (this wasn't even a consideration when we developed that functionality) and more to do with the approval process in certain countries.

Also, does Valve or Sierra Studios keep track of how many total players participate in multi-player, death-match type games?

We have statistics for the HL engine, but not great statistics for other publishers'/developers' stuff. In the first year after shipping HL (after the first holiday season), the number of online players grew 5x, in the second year, it grew 10x. This year, it is on pace to grow 10x again. Almost every week we see an all-time high in number of simultaneous players and aggregate play time. We see over 1.7 billion aggregate minutes of play per month (which is more view minutes than most top 10 or 20 TV shows) and we see over 65000 users playing at any one time. We are an order of magnitude larger in these statistics than any other action title in our space.

It's definitely one of the more popular death-match games around, but does it keep attracting new players?

Yes, see above.

Finally, although this might be a hard question to answer, what do you think in a nutshell is the multi-player mode's impact on death-match games and on gaming in general?

All of our multiplayer modes (even HLDM really) have focused on what we believed was key to the multiplayer experience, the social interaction between players. That's why we think games like TFC and Counter Strike have really caught on. DM in its purest form is, in our opinion, not as interesting because it's pretty repetitive. That's not to say that there aren't some really fun pure DM games.

Interview with Ken Birdwell

Q: To start, what inspired your team to make a 1st-person shooter?

A: We originally started the company - august 1996 - to create a massively multiplayer first person strategy combat game - something like a first person C&C - but we were nervous about doing a game with too many technical unknowns as our first project. After talking it over for a bit, since Mike Harrington (one of the founders) know Mike Abrash (at id) pretty well, we figured we had a good line of getting a Quake 1 license so we decided to make a first person shooter because we could do it without writing a lot of code and not taking too long (don't laugh).

Q: What were your feelings about plunging into a saturated genre such as this?

A: It was tough. We knew Quake 2 was coming, and there wasn't much we could do to compete with that, but there were also the "Hot-hot-hot" games like Diakatana, SiN, Prey, Amen, Prax War, and a bunch of others coming out that were the next "Quake Killer" and we were afraid we'd just get lost in the coming crush. Initial press on Half-Life was of the form "Half-Life: I guess they couldn't afford a full one!" and the like. Not good. We then went on a 9 month crunch of adding new core technology so that we could compete at least on features with the other games, features that our competitors ended up cutting. In the end, most all of our competitors either canceled or head-planted really badly, so in the end we were pretty much the only game in our genre to actually ship. My basic attitude is all magazine previews are nonsense, even ours, so go ahead and ignore them. If your game is good, it'll sell. If it's not, it won't. If two good games come out at the same time, people will buy both. If no games are any good, nothing will sell. I figure word-of-mouth accounts for over 90% of all sales, so regardless of how good you look in print, if your play sucks you won't sell, at least not for long.

Q: When did your team start its work on the game, and how long did it take before the team felt like it was finally on the "right track" ?

A: Over a year. You can find the process we undertook to go from being a mediocre game to a top 10 in my GDMag article. Other than simplifying the chronology - the real story is a lot messier - it's pretty accurate.

Q: Were there any particular business motivations that pushed or forced the design process?

A: By late 97, most of the core technology was done, but the game still sucked. A number of other competing games had come out, but it became really clear that only the top 10 games (not just percent, but top 10) made any money. Given the amount of time and effort even a bad game would take, none of the core people at Valve wanted to waste their time on anything other than a being a top 10.

Q: Were there additions or subtractions from the game caused by business concerns, such as being too difficult or too easy, or requiring too much computer processing power?

A: To be a top 10, we needed to hit our core demographics, which are male, casual to moderate gamers 13 to 35(?). Valve is made up of an odd mix of hardcore and non gamers, so we tuned the game primarily through playtest sessions - about 120 or so - of people in our target group. From these 1-2 hour sessions we would judge the enjoyment/frustration levels during each section and edited the game based on the results.

Our game also needed to run on the majority of computers owned by people buying games. These are typically low end computers bought that year, along with all the high end computers bought within about a year. With a ship date of sometime in 1998, this was about a P-200 with 32MB of ram and a Voodoo 1 graphics card on up. Artwork was scaled to fit those performance characteristics.

Q: How was the game received by critics of games and of violence in games?

A: Critics loved us from the 1998 GDC on. Getting critics to understand what we're trying to do is what Gabe Newell is better at than anyone else in the industry. Violence critics didn't get much airplay until well after we shipped, and by then there were other more obviously gratuitously violent games available so our PG-13 action-movie level of violence makes an odd, and easily defensible, choice to be attacked.

Q: I read in another article that Valve licensed id Software's Quake 2 engine as part of the design process for "Half-Life," but I also read on the game's official site that a lot of the AI and map designing technologies for the game were invented by your team. Are other companies and game developers now licensing your technologies, too, to take advantage of them?

A: We're Quake 1, not Quake 2. We have a handful of lines from Quake 2 in our game to fix some old Quake 1 bugs, but other than that we're either Quake 1 or our own code. We replaced about 70% of the code, just about everything except the core renderer and architecture, and about half the tools. All the AI, the Map editor "Worldcraft", client interface, and the networking are all ours, but the BSP, visibility, span fillers, and world renderers are mostly untouched Quake 1 code.

The main people taking advantage of our technology are Mod makers, and we've been really happy with their results. Very few other companies have the time to learn what it is we have, and to be honest we're not very aggressive about licensing it. It took us about a year to understand the id code base, and we added about that much again on top of it. This is typically too much to learn in the time frame of shipping a project so most companies feel it's better to create the technology themselves so that their people understand

it than it is to buy it. We disagree, and we'll continue to license code when it makes sense, but a lot of people feel differently and I understand why they do.

Q: Finally, although this might be a hard question to answer, what do you think in a nutshell is "Half-Life's" overall impact on 1st-person shooters and on gaming in general?

A: As far as 1st person shooter impact goes, I don't know. I'd like to think that we helped make games like NOLF possible, but any real influence is probably more political instead of artistic or technical. We've heard a number of anecdotal stories about development groups going to upper management and asking for more time or to be able to release a patch or do play testing or something basic like that and being denied, but then using the "Valve did it this way" argument to successfully get approval to do the right thing! There are a lot of talented people who could have made great games, but are crushed by silly decisions in upper management. If we've given people a way to fight that, then I think we've done a good thing.

CMP Game Media Group Presents:

Gamasutra.com

Gamasutra

The Art and Science of Making Games

The Cabal: Valve's Design Process For Creating **Half-Life**

By **Ken Birdwell**

Gamasutra

December 10, 1999

URL: http://www.gamasutra.com/features/19991210/birdwell_01.htm

While *Half-Life* has seen resounding critical and financial success (winning over 50 Game of the Year awards and selling more than a million copies worldwide), few people realize that it didn't start out a winner — in fact, Valve's first attempt at the game had to be scrapped. It was mediocre at best, and suffered from the typical problems that plague far too many games. This article is about the teamwork — or "Cabal process" — that turned our initial, less than impressive version of *Half-Life* into a groundbreaking success.

Paving the Way with Good Intentions

Our initial target release date was November 1997 — a year before the game actually shipped. This date would have given Valve a year to develop what was in essence a fancy *Quake TC* (Total Conversion — all new artwork, all new levels). By late September 1997, nearing the end of our original schedule, a whole lot of work had been done, but there was one major problem — the game wasn't any fun.

Yes, we had some cool monsters, but if you didn't fight them exactly the way we had planned they did really stupid things. We had some cool levels, but they didn't fit together well. We had some cool technology, but for the most part it only showed up in one or two spots. So you couldn't play the game all the way through, none of the levels tied together well, and there were serious technical problems with most of the game. There were some really wonderful individual pieces, but as a whole the game just wasn't working.

The obvious answer was to work a few more months, gloss over the worst of the problems and ship what we had. For companies who live and die at the whim of their publishers, this is usually the route taken — with predictable results. Since Valve is fairly independent, and since none of us believed that we were getting any closer to making a game we could all like, we couldn't see how a month or two would make any significant difference. At this point we had to make a very painful decision — we decided to start over and rework every stage of the game.



Many of our scripted sequences were designed to give the player game-play clues as well as provide moments of sheer terror.

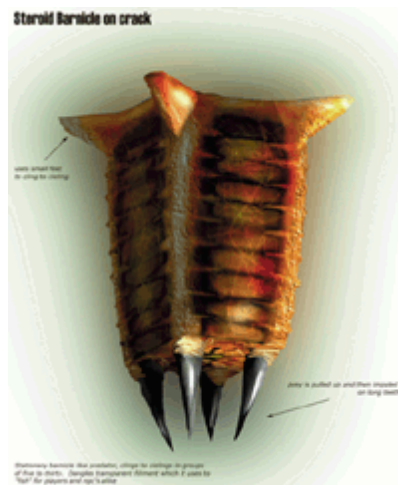
Fortunately, the game had some things in it we liked. We set up a small group of people to take every silly idea, every cool trick, everything interesting that existed in any kind of working state somewhere in the game and put them into a single prototype level. When the level started to get fun, they added more variations of the fun things. If an idea wasn't fun, they cut it. When they needed a software feature, they simplified it until it was something that could be written in a few days. They all worked together on this one small level for a month while the rest of us basically did nothing. When they were done, we all played it. It was great. It was *Die Hard* meets *Evil Dead*. It was the vision. It was going to be our game. It was huge and scary and going to take a lot of work, but after seeing it we weren't going to be satisfied with anything less. All that we needed to do was to create about 100 more levels that were just as fun. No problem.

So, Tell Me About Your Childhood

The second step in the pre-cabal process was to analyze what was fun about our prototype level. The first theory we came up with was the theory of "experiential density" — the amount of "things" that happen to and are done by the player per unit of time and area of a map. Our goal was that, once active, the player never had to wait too long before the next stimulus, be it monster, special effect, plot point, action sequence, and so on. Since we couldn't really bring all these experiences to the player (a relentless series of them would just get tedious), all content is distance based, not time based, and no activities are started outside the player's control. If the players are in the mood for more action, all they need to do is move forward and within a few seconds something will happen.

The second theory we came up with is the theory of player acknowledgment. This means that the game world must acknowledge players every time they perform an action. For example, if they shoot their gun, the world needs to acknowledge it with something more permanent than just a sound — there should be some visual evidence that they've just fired their gun. We would have liked to put a hole through the wall, but for technical and game flow reasons we really couldn't do it. Instead we decided on "decals" — bullet nicks and explosion marks on all the surfaces, which serve as permanent records of the action. This also means that if the player pushes on something that should be pushable, the object shouldn't ignore them, it should move. If they whack on something with their crowbar that looks like it should break, it had better break. If they walk into a room with other characters, those characters should acknowledge them by at least looking at them, if not calling out their name. Our basic theory was that if the world ignores the player, the player won't care about the world.

A final theory was that the players should always blame themselves for failure. If the game kills them off with no warning, then players blame the game and start to dislike it. But if the game hints that danger is imminent, show players a way out and they die anyway, then they'll consider it a failure on their part; they've let the game down and they need to try a little harder. When they succeed, and the game rewards them with a little treat — scripted sequence, special effect, and so on — they'll feel good about themselves and about the game.



Conceptual artwork for ceiling-mounted monster that was dangerous to both the player and the player's enemies

Secret Societies

Throughout the first 11 months of the project we searched for an official "game designer," — someone who could show up and make it all come together. We looked at hundreds of resumes and interviewed a lot of promising applicants, but no one we looked at had enough of the qualities we wanted for us to seriously consider them the overall godlike "game designer" that we were told we needed. In the end, we came to the conclusion that this ideal person didn't actually exist. Instead, we would create our own ideal by combining the strengths of a cross section of the company, putting them together in a group we called the "Cabal."

The goal of this group was to create a complete document that detailed all the levels and described major monster interactions, special effects, plot devices, and design standards. The Cabal was to work out when and how every monster, weapon, and NPC was to be introduced, what skills we expected the player to have, and how we were going to teach them those skills. As daunting as that sounds, this is exactly what we did. We consider the Cabal process to have been wildly successful, and one of the key reasons for Half-Life's success.

Cabal meetings were semi-structured brainstorming sessions usually dedicated to a specific area of the game. During each session, one person was assigned the job of recording and writing up the design, and another was assigned to draw pictures explaining the layout and other details. A Cabal session would typically consist of a few days coming up with a mix of high level concepts for the given area, as well as specific events that sounded fun.



The team explored a variety of visual metaphors that resulted in some very unique and effective opponents.

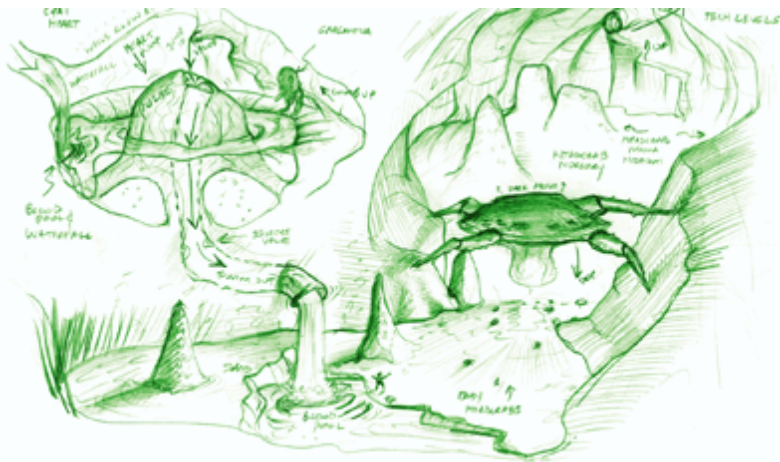
Once enough ideas were generated, they would be reorganized into a rough storyline and chronology. Once this was all worked out, a description and rough sketch of the geometry would be created and labeled with all the key events and where they should take place. We knew what we wanted for some areas of the game from the very start, but other areas stayed as "outdoors" or "something with a big monster" for quite some time. Other areas were created without a specific spot in the game. These designs would sit in limbo for a few weeks until either it became clear that they weren't going to fit, or that perhaps they would make a good segue between two other areas. Other portions were created to highlight a specific technology feature, or simply to give the game a reason to include a cool piece of geometry that had been created during a pre-cabal experiment. Oddly enough, when trying to match these artificial constants, we would often create our best work. We eventually got into the habit of placing a number of unrelated requirements into each area then doing our best to come up with a rational way to fit them together. Often, by the end of the session we would find that the initial idea wasn't nearly as interesting as all the pieces we built around it, and the structure we had designed to explain it actually worked better without that initial idea.

During Cabal sessions, everyone contributed but we found that not everyone contributed everyday. The meetings were grueling, and we came to almost expect that about half of the group would find themselves sitting through two or three meetings with no ideas at all, then suddenly see a direction that no one else saw and be the main contributor for the remainder of the week. Why this happened was unclear, but it became important to have at least five or six people in each meeting so that the meetings wouldn't stall out from lack of input.

The Cabal met four days a week, six hours a day for five months straight, and then on and off until the end of the project. The meetings were only six hours a day, because after six hours everyone was emotionally and physically drained. The people involved weren't really able to do any other work during that time, other than read e-mail and write up their daily notes.

The initial Cabal group consisted of three engineers, a level designer, a writer, and an animator. This represented all the major groups at Valve and all aspects of the project and was initially weighted towards people with the most product experience (though not necessarily game experience). The Cabal consisted only of people that had actual shipping components in the game; there

were no dedicated designers. Every member of the Cabal was someone with the responsibility of actually doing the work that their design specified, or at least had the ability to do it if need be.



It's important to include information on the intended path through the level, as well as rough geometry and character placement.

The first few months of the Cabal process were somewhat nerve wracking for those outside the process. It wasn't clear that egos could be suppressed enough to get anything done, or that a vision of the game filtered through a large number of people would be anything other than bland. As it turned out, the opposite was true; the people involved were tired of working in isolation and were energized by the collaborative process, and the resulting designs had a consistent level of polish and depth that hadn't been seen before.

Internally, once the success of the Cabal process was obvious, mini-Cabals were formed to come up with answers to a variety of design problems. These mini-Cabals would typically include people most effected by the decision, as well as try to include people completely outside the problem being addressed in order to keep a fresh perspective on things. We also kept membership in the initial Cabal somewhat flexible and we quickly started to rotate people through the process every month or so, always including a few people from the last time, and always making sure we had a cross section of the company. This helped to prevent burn out, and ensured that everyone involved in the process had experience using the results of Cabal decisions.

The final result was a document of more than 200 pages detailing everything in the game from how high buttons should be to what time of the day it was in any given level. It included rough drawings of all the levels, as well as work items listing any new technology, sounds, or animations that those levels would require.

We also ended up assigning one person to follow the entire story line and to maintain the entire document. With a design as large as a 30-hour movie, we ended up creating more detail than could be dealt with on a casual or part-time basis. We found that having a professional writer on staff was key to this process. Besides being able to add personality to all our characters, his ability to keep track of thematic structures, plot twists, pacing, and consistency was invaluable.

Pearls Before Swine

By the second month of the Cabal, we (the "swine") had enough of the game design to begin development on several areas. By the third month, we had enough put together to begin play testing.

A play-test session consists of one outside volunteer (Sierra, our publisher, pulled play-testers from local people who had sent in product registration cards for other games) playing the game for two hours. Sitting immediately behind them would be one person from the Cabal session that worked on that area of the game, as well as the level designer who was currently the "primary" on the level being tested. Occasionally, this would also include an engineer if new AI needed to be tested.

Other than starting the game for them and resetting it if it crashed, the observers from Valve were not allowed to say anything. They had to sit there quietly taking notes, and were not allowed to give any hints or suggestions. Nothing is quite so humbling as being forced to watch in silence as some poor play-tester stumbles around your level for 20 minutes, unable to figure out the "obvious" answer that you now realize is completely arbitrary and impossible to figure out.

This was also a sure way to settle any design arguments. It became obvious that any personal opinion you had given really didn't mean anything, at least not until the next play-test session. Just because you were sure something was going to be fun didn't make it so; the play-testers could still show up and demonstrate just how wrong you really were.

A typical two-hour play-test session would result in 100 or so "action items" — things that needed to be fixed, changed, added, or deleted from the game. The first 20 or 30 play-test sessions were absolutely critical for teaching us as a company what elements were fun and what elements were not. Over the course of the project we ended up doing more than 200 play-test sessions, about half of them with repeat players. The feedback from the sessions was worked back into the Cabal process, allowing us to preemptively remove designs that didn't work well, as well as elaborate on designs that did.

Toward the middle of the project, once the major elements were in place and the game could be played most of the way through, it became mostly a matter of fine-tuning. To do this, we added basic instrumentation to the game, automatically recording the player's position, health, weapons, time, and any major activities such as saving the game, dying, being hurt, solving a puzzle, fighting a monster, and so on. We then took the results from a number of sessions and graphed them together to find any areas

where there were problems. These included areas where the player spent too long without any encounters (boring), too long with too much health (too easy), too long with too little health (too hard), all of which gave us a good idea as to where they were likely to die and which positions would be best for adding goodies.



Letting players see other characters make mistakes that they'll need to avoid is an effective way to explain your puzzles and add tension and entertainment value.



This creature was initially designed as a friendly character, but play-testing revealed players' tendencies to shoot first and ask questions later.

Another thing that helped with debugging was making the "save game" format compatible between the different versions of the engine. Since we automatically saved the game at regular intervals, if the play-testers crashed the game we would usually have something not too far from where they encountered the bug. Since these files would even work if the code base they were testing was several versions old, it made normally rare and hard to duplicate bugs relatively easy to find and fix. Our save game format allowed us to add data, delete data, add and delete code (we even supported function pointers) at will, without breaking anything. This also allowed us to make some fairly major changes after we shipped the game without interfering with any of our players' hard-won saved games.

No Good Deed Goes Unpunished

Until the Cabal process got underway, technology was added to Half-Life freely. It was assumed that "if we build it, they will come," meaning that any new technology would just naturally find a creative use by the content creation folks. A prime example of this fallacy was our "beam" effect, basically a technique for doing highly tunable squiggly glowing lines between two points; stuff like lightning, lasers, and mysterious glowing beams of energy. It was added to the engine, the parameters were exposed, and an e-mail was sent out explaining it. The result was ... nothing. After two months only one level designer had put it in a map. Engineering was baffled.

During the Cabal process, we realized that although the level designers knew of the feature, they really had no clear idea of what it was for. The parameters were all very cryptic, and the wrong combinations would cause the beams to have very ugly-looking effects. There were no decent textures to apply to them, and setting them up was a bit of a mystery. It became very clear the technology itself was only a small part of the work and integration, training, and follow-through were absolutely necessary to make the technology useful to the game. Writing the code was typically less than half the problem.

Square Pegs

Practically speaking, not everyone is suited for the kind of group design activity we performed in the Cabal, at least not initially. People with strong personalities, people with poor verbal skills, or people who just don't like creating in a group setting shouldn't be forced into it. We weighted our groups heavily toward people with a lot of group design experience, well ahead of game design experience. Even so, in the end almost everyone was in a Cabal of one sort or another, and as we got more comfortable with this process and started getting really good results it was easier to integrate the more reluctant members. For current projects, such as Team Fortress 2, the Cabal groups are made up of 12 or more people, and rarely fewer than eight. The meetings ended up being shorter, and they also ended up spreading ideas around a lot quicker, but I'm not sure I'd recommend that size of group initially.

Just about everything in Half-Life was designed by a Cabal. This at first seemed to add a bit of overhead to everything, but it had the important characteristic of getting everyone involved in the creation process who were personally invested in the design. Once everyone becomes invested in the design as a whole, it stops being separate pieces owned by a single person and instead the entire game design becomes "ours."

This "ours" idea extended to all levels. Almost every level in the game ended up being edited by at least three different level designers at some point in its development and some levels were touched by everyone. Though all the level designers were good at almost everything, each found they enjoyed some aspect of level design more than other aspects. One would do the geometry, one would do monster and AI placement, our texture artist would step in and do a texturing pass, and then one would finish up with a lighting pass, often switching roles when needed due to scheduling conflicts. This became critical toward the end of the project when people finished at different times. If a play-test session revealed something that needed to be changed, any available level designer could make the changes without the game getting bottlenecked by needing any specific individual.



By placing traditional combat action in more challenging environments we were able to intensify the feeling of tension and suspense.

This idea also extended to all code, textures, models, animations, sounds, and so on. All were under source control and any individual was able to synch up to the sources and make whatever changes were necessary. With a little bit of self-control, this isn't as random as it sounds. It had the added benefit in that it was fairly easy to get a daily record of exactly what was changed and by whom. We would then feed this information back into the play-test cycles, only testing what had changed, as well as helping project scheduling by being able to monitor the changes and get a pretty good estimate of the stability and completeness of any one component. This also allowed us to systematically add features throughout the process with minimal impact. Once the technical portion was completed, the engineer assigned to the feature was able to synch to all the source artwork and rebuild any and all files (models, textures, levels, and so on) affected by the change.

The Workers Control the Means of Production

Even with all emphasis on group activity, most of the major features of Half-Life still only happened through individual initiative. Everyone had different ideas as to what exactly the game should look like, or at least what features we just had to do. The Cabal process gave these ideas a place to be heard, and since it was accepted that design ideas can come from anyone, it gave people as much authority as they wanted to take. If the idea required someone other than the inventor to actually do the work, or if the idea had impact on other areas of the game, they would need to start a Cabal and try to convince the other key people involved that their idea was worth the effort. At the start of the project, this was pretty easy as most everyone wildly underestimated the total amount of work that needed to be done, but toward the middle and end of the project the more disruptive decisions tended to get harder and harder to push through. It also helped filter out all design changes except for the ones with the most player impact for the least development work.



Placing the player in a soldier-vs.-alien conflict helped reinforce the illusion of an active environment, and let Valve show off its combat AI with minimal risk to the player.

Through constant cycle of play-testing, feedback, review, and editing, the Cabal process was also key in removing portions of the game that didn't meet the quality standards we wanted, regardless of the level of emotional attachment the specific creator may have had to the work. This was one of the more initially contentious aspects of the Cabal process, but perhaps one of the more important. By its very nature, the Cabal process avoided most of the personal conflicts inherent in other more hierarchical organizations. Since problems were identified in a relatively objective manner of play-testing, and since their solutions were arrived at by consensus or at least by an individual peer, then an authority that everyone could rebel against just didn't exist.

On a day-to-day basis, the level of detail supplied in even a 200-page design document is vague at best. It doesn't answer the 1,001 specific details that each area requires, or the countless creative details that are part of everyday development. Any design document is really nothing more than a framework to work from and something to improve the likelihood that work from multiple people will fit together in a seamless fashion. It's the Cabal process that helped spread around all the big picture ideas that didn't make it into any document — things that are critical to the feel of the game, but too nebulous to put into words. It also helps maximize individual strengths and minimize individual weaknesses and sets up a framework that allows individuals to influence as much of the game as possible. In Half-Life, it was the rare area of the game that didn't include the direct work of more than ten different people, usually all within the same frame.

In order for highly hierarchical organizations to be effective, they require one person who understands everyone else's work at least as well as the individuals doing the work, and other people who are willing to be subordinates yet are still good enough to actually implement the design. Given the complexity of most top game titles, this just isn't practical — if you were good enough to do the job, why would you want to be a flunky? On the other hand, completely unstructured organizations suffer from lack of information and control — if everyone just does their own thing, the odds that it'll all fit together in the end are somewhere around zero.

At Valve, we're very happy with the results of our Cabal process. Of course, we still suffer from being overly ambitious and having, at times, wildly unrealistic expectations, but these eventually get straightened out and the Cabal process is very good about coming up with the optimal compromise. Given how badly we failed initially, and how much the final game exceeded our

individual expectations, even our most initially reluctant person is now a staunch supporter of the process.

Tips for a Successful Cabal

- Include an expert from every functional area (programming, art, and so on). Arguing over an issue that no one at the meeting actually understands is a sure way to waste everyone's time.
- Write down everything. Brainstorming is fine during the meetings, but unless it's all written down, your best ideas will be forgotten within days. The goal is to end up with a document that captures as much as is reasonable about your game, and more importantly answers questions about what people need to work on.



- Not all ideas are good. These include yours. If you have a "great idea" that everyone thinks is stupid, don't push it. The others will also have stupid ideas. If you're pushy about yours, they'll be pushy about theirs and you're just going to get into an impasse. If the idea is really good, maybe it's just in the wrong place. Bring it up later. You're going to be designing about 30 hours of game play; if you really want it in it'll probably fit somewhere else. Maybe they'll like it next month.
- Only plan for technical things that either already work, or that you're sure will work within a reasonable time before play testing. Don't count on anything that won't be ready until just before you ship. Yes, it's fun to dream about cool technology, but there's no point in designing the game around elements that may never be finished, or not polished enough to ship. If it's not going to happen, get rid of it, the earlier the better.
- Avoid all one-shot technical elements. Anything that requires engineering work must be used in more than one spot in the game. Engineers are really slow. It takes them months to get anything done. If what they do is only used once, it's a waste of a limited resource. Their main goal should always be to create tools and features that can be used everywhere. If they can spend a month and make everyone more productive, then it's a win. If they spend a week for ten seconds of game play, it's a waste.

Ken is senior developer at Valve and has contributed to a wide range of projects in the last 15 years, most recently on animation and AI for **Half-Life**. Previous projects include satellite networking, cryptography, 3D prosthetic design tools, 3D surface reconstruction, and in-circuit emulators. Oddly enough, Ken dropped out of studying EE to pursue a fine arts degree at The Evergreen State College, which he considers far more relevant to creative thinking than any silly differential equations class. You can reach him at kenb@valvesoftware.com.

Copyright © 2000-2001 CMP Media Inc. All rights reserved.

planet

HALF-LIFE

Half-Life



THE SHACK PROJECT

[BZII News](#) - [Files](#) - [Fixes](#) - [Mods](#) - [More](#)

[PlanetHalf-Life](#) | [Half-Life](#) | [Half-Life Guide](#)

News

[Current](#) / [Submit](#)
[Archive](#) / [Search](#)
[POTD](#) / [Submit](#)

Files

[Half-Life Files](#)

Community

[Hosted Sites](#)
[Get Hosted!](#)
[Forums](#)
[Chat](#)
[Contact Us](#)
[Advertise With Us](#)
[PHL Gallery](#)
[PHL Staff](#)
[Help Wanted](#)
[Links](#)

Half-Life Guide

[Overview](#)
[FAQ](#)
[Allies](#)
[Enemies](#)
[Weapons](#)
[Console Commands](#)
[Walkthrough](#)

Multiplayer

[Overview](#)
[GameSpy Arcade](#)
[GameSpy3D](#)
[Using Mods](#)

Add Ons

[Modifications](#)
[Levels](#)
[Models/Skins](#)
[Editing](#)
[Resources](#)

 breaking news from


GameSpy Network

[GameSpy](#)
[GameSpy.com](#)
[GameSpyDaily](#)
[GameSpy 3D](#)
[GameSpy Arcade](#)

Half-Life: The PlanetHalfLife Guide

The Half-Life Story

The majority of Half-Life's storyline was written by Marc Laidlaw, Valve Software's resident wordsmith and author of novels such as Dad's Nuke, Kalifornia, and The 37th Mandala.

Deep in the bowels of the Black Mesa Research Labs, a decommissioned missile base, a top secret project is underway. Information about the project is strictly on a "need-to-know" basis, and as a low level research associate you (Gordon Freeman) "need to know" very little. Each morning you ride the train to work from the employee dorms, you put on your environmental protection suit, you enter the test chamber, and you run stress tests on whatever odd devices have been delivered from some other nameless part of the Black Mesa compound.

But this morning is different. This morning, your test lab is suddenly the most important place on Earth--because something is going seriously wrong. Maybe it's sabotage--maybe it's an accident. Whatever the reason, reality is getting all bent out of shape. One minute you're doing your job, pressing buttons. The next thing you know, you're staring into an alien world. Something huge with too many arms is taking a bite out of your partner's face. An explosion of unearthly light....then darkness.

Disaster. Sirens wailing. People screaming. And everywhere you turn, people are dying--being eaten. Monsters are everywhere. Monsters--there's no better word for them. You head fro the surface, to get the hell away from ground zero, but the usual routes are unpassable--damaged by the disaster, infested with headcrabs and houndeyes and increasingly larger and hungrier creatures. Madness is the order of the day. You enlist the help of traumatized scientists and trigger-happy guards to get through high security zones, sneaking and fighting your way through riuned missile silos and Cold War cafeterias, through darkened air ducts and subterranean railways where you must ride a missile transport sled straight into the jaws of slaving nightmare. When you finally come in sight of the surface, you realize the aliens aren't your only enemies--for now the government forces have arrived with heavy-weapons goons, squadrons of ruthless containment troops, and stealthy assassin gals. Their orders seem to be that when it comes to Black Mesa labs, nothing must get out alive....and especially not you, the guy who made it all go bad. So much for the cavalry.

When your own species turns against you, where do you turn? You've uprooted a bunch of nasty government secrets. You've found a portal to another world, and an alien light comes shining through. Can it get any worse over there? Some things you just have to see for yourself.

Gordon Freeman

In Half-Life, you play Gordon Freeman. A native of Seattle, Washington, Gordon Freeman showed high interest and aptitude in the areas of quantum physics and relativity at an extremely young age. His earliest heroes were Einstein, Hawking and Feynman.

While a visiting student at the University of Innsbruck in the late 1990's, Gordon Freeman observed a series of seminal teleportation experiments conducted by the Institute for

[GameLaunch 3D](#)[GameSpy_Store](#)

Utilities

[CheatingPlanet](#)[FilePlanet](#)[GameSpyder](#)

Action:

[3DActionPlanet](#)[Captured](#)[PlanetAllegiance](#)[PlanetAvP](#)[PlanetBlood](#)[PlanetDaikatana](#)[PlanetDescent](#)[PlanetDrakan](#)[PlanetDuke](#)[PlanetFortress](#)[PlanetHalfLife](#)[PlanetHeretic](#)[PlanetKingpin](#)[PlanetQuake](#)[PlanetShogo](#)[PlanetSoldier](#)[PlanetStarsiege](#)[PlanetUnreal](#)[PlanetWheelofTime](#)

RPG:

[RPGPlanet](#)[PlanetAnachronox](#)[PlanetBaldursGate](#)[PlanetDeusEx](#)[PlanetDiablo](#)[PlanetEverQuest](#)[PlanetVampire](#)

Sports:

[SportPlanet](#)[PlanetDeerHunter](#)

Strategy:

[StrategyPlanet](#)[PlanetAgeofEmpires](#)[PlanetAnnihilation](#)[PlanetBattlezone](#)[PlanetBlackAndWhite](#)[PlanetCNC](#)

Tactical:

[PlanetRainbow6](#)[TacticalPlanet](#)

Classic/Console:

[ClassicGaming](#)[PlanetDreamcast](#)[PlanetPS2](#)[PlanetXbox](#)

Hardware:

[PlanetHardware](#)

Music:

[RadioSpy](#)

Community:

[Battlegrounds](#)[GameArt](#)[LANParty.com](#)[GameFinger](#)[Stormtroopers](#)

Experimental Physics (see Bowemeester, Pan, Mattle, Eibl, Weinfurter, Zeilinger, "Experimental Quantum Teleportation," Nature, 11 December 1997) (see also <http://www.sciam.com/explorations/122297/teleport>).

Practical applications for teleportation became his obsession. In 1999, Freeman received his doctorate from M.I.T. with a thesis paper entitled: "Observation of Einstein-Podolsky-Rosen Entanglement on Supraquantum Structures By Induction through Nonlinear Transuranic Crystal of Extremely Long Wavelength (ELW) Pulse from Mode-Locked Source Array."



Disappointed with the slow pace and poor funding of academic research, and with tenure a distant dream, Gordon cast about for a job in private industry. As fortune would have it, his mentor at M.I.T., Professor Alex Kleiner, had taken charge of a research project being conducted at a decommissioned missile base in Black Mesa, New Mexico. Kleiner was looking for a few bright associates, and Gordon was his first choice. Considering the source and amount of funds available to the Black Mesa Labs, Gordon suspected that he would be involved in some sort of weapons research; but in the hopes that practical civilian applications would arise (in areas of quantum computing and astrophysics), he accepted Kleiner's offer. Apart from a butane-powered tennis ball cannon he constructed at age 6, Gordon had never handled a weapon of any sort-or needed to... until now.

The Half-Life Technology

Half-Life is based on the Quake(tm) engine by ID Software, with Valve's own enhancements to the engine, such as 16-bit and 24-bit color and MMX support, as well as being developed to take full advantage of 3dfx's Voodoo2. Half-Life is based on a whole new level of proprietary technology creating a extremely rich and original gaming experience.

- Rendering

So you don't want to have to buy a special hardware accelerator just to get 16-bit color, colored lighting, blurring, translucency or other cool visual effects? Then don't. Half-Life has developed all these features in software so now they're an integral part of the game play, not just eye-candy. Of course, if you do have Open-GL, Direct 3D or MMX hardware, things will look mind-bogglingly cool.

- Skeletal Animation System

Hand-in-glove with a demand for realistic lighting and color effects is a desire for monsters that look and move as realistically as possible. To accomplish this goal, the engineers at Valve have created a skeletal animation system for monsters. Rather than store a discrete set of polygonal meshes for each key frame of animation, as traditional action games do, the skeletal system moves the "bones" within a monster and deforms a mesh and texture map around them. There are a number of advantages this gives Half-Life animators as they build more compelling and complex monsters: Smoother and richer animation Half-Life players will see much smoother animation than in typical action games. While both sprite- and mesh-based animation systems are based on a fixed keyframe animation rate, which is typically targeted at the lowest common denominator system, Half-Life's skeletal animation system does not limit the number of frames in an animation. For instance, a typical walk cycle may have as many as 80 frames in Half-Life, as compared to only 4 in some sprite-based games.

- Monster AI

Half-Life's monsters and life-forms are also remarkably--even terrifyingly--intelligent. Valve has created a technology that imbues Half-Life monsters with tactical intelligence, multi-character cooperation, and a supreme will to live. The result is a menagerie of new creatures whose intelligence and unpredictability make them truly formidable adversaries. Traditionally, game AI is a set of hard-coded if-then decisions for every possible situation that could confront a monster, such as, "If there is a bad guy in this room then shoot at him." Valve took another tack, designing a module-based AI system that provides practically infinite flexibility and monster growth potential.

- Decal System

With Dynamically changing surfaces/Decals Surfaces in Half-Life are dynamic. They can change over time or as the player interacts with them. Damp walls

may grow mossy, water will ripple as the player moves through it and, through the use of "decal" technology, hard surfaces will retain the scars of a previous firefight. Decals--spot painting effects over existing textures--also make it possible for opponents to leave threatening graffiti on walls, or for blood, water and smoke to leave their marks on both surfaces and characters in the game.

- Real-Time DSP

The sound in Half-Life is astounding due to DSP sound which calculates the direction of a sound and the size and material a room is made of to alter that sound accordingly to fit the area, a gunshot outside will sound different than a gunshot in a metal room or underwater. This also saves disk space since the sound is being altered over and over instead of many sounds that hardly get used. This is sure to make your ears smile.

System Requirements

Requirements:

CPU Requirements
CD-ROM
Memory Requirements
O/S Requirements
Monitor/Video Card
3D Acceleration Support
Sound Card Support
Controllers

Minimum

- Pentium 100 or better
- 2X CD-ROM
- 16 MB RAM
- Windows '95
- SVGA 640 x 480, 16-bit
Not Required
- Windows compatible card
- Mouse, Keyboard

Recommended

- Pentium 166 MMX
- 4X CD-ROM or better
- 32 MB RAM preferred
- Windows '95
- SVGA 640 x 480, 16-bit
- 3D accelerator recommended
- Sound-Blaster compatible
- Joystick or Gamepad optional



[\[Main Page\]](#) [\[Half-Life\]](#) [\[Opposing Force\]](#) [\[Counter-strike\]](#) [\[TFC\]](#) [\[Hosted Sites\]](#)

© 1996-2001 GameSpy Industries. [Contact us](#) for more information on GameSpy Industries.
Be sure to read our and check out how you can and target your products and services to gamers.

H A L F - L I F E

CLICK TO ENTER...