

Postmortem: Blizzard's *Diablo II*

By Erich Schaefer

The original *Diablo* went gold on the day after Christmas in 1996, after a grueling four-month crunch period. We hadn't put any thought into what game to do next, but as most developers can probably relate to, we were pretty certain we weren't ready to return to the *Diablo* world after such a long development cycle. The only thing we were certain of was that we wanted to avoid another crunch like we had just experienced. *Diablo II* went gold on June 15, 2000, after a grueling 12-month crunch period.

After *Diablo* shipped, we spent about three months recovering and kicking around game ideas for our next project, but nothing really stuck. The idea of returning to *Diablo* began to creep into the discussions, and after a couple of months of recuperation, we suddenly realized we weren't burned out on *Diablo* anymore. We dusted off the reams of wish-list items we had remaining from the original, compiled criticisms from reviews and customers, and began brainstorming how we could make *Diablo II* bigger and better in every way.

Diablo II never had an official, complete design document. Of course, we had a rough plan, but for the most part we just started off making up new stuff: four towns instead of the original game's one; five character classes, all different from the previous three; and many new dungeons, vast wilderness tile-sets, and greatly expanded lists of items, magic, and skills. We wanted to improve upon every aspect of the original. Where *Diablo* had three different armor "looks" for each character, *Diablo II* would use a component system to generate hundreds of variations. Where *Diablo* had "unique" boss monsters with special abilities, *Diablo II* would have a system for randomly generating thousands of them. We would improve the graphics with true transparency, colored light sources, and a quasi-3D perspective mode. Level loads would be a thing of the past. The story would be factored in from the beginning and actually have some bearing on the quests. We knew creating this opus would be a big job. Because we had the gameplay basics already polished, we figured we would hire some new employees, create some good tools, and essentially make four times the original game doing only two times the work. We estimated a two-year development schedule.



While the player characters are only seen in the game as 75 pixels tall, all were modeled and rendered in high resolution for use on the character selection screen and in promotional materials. Here, the Paladin stands tall.

The *Diablo II* team comprised three main groups: programming, character art (everything that moves), and background art (everything that doesn't move), with roughly a dozen members each. Design was a largely open process, with members of all teams contributing. Blizzard Irvine helped out with network code and Battle.net support. The Blizzard film department (also in Irvine) contributed the cinematic sequences that bracket each of *Diablo*'s acts, and collaborated on the story line.

Almost all of *Diablo II*'s in-game and cinematic art was constructed and rendered in 3D Studio Max, while textures and 2D interface elements were created primarily with Photoshop. The programmers wrote in C and some C++, using Visual Studio and SourceSafe for version control.



Creating detailed sketches of settings, such as this hut in the Act III dock town of Kurast, preceded the actual modeling of background art.

Blizzard North started out as Condor Games in September 1993. The first contracts we landed were ports of Acclaim's Quarterback Club football games for handheld systems and, more significantly, a Sega Genesis version of Justice League Task Force for Sunsoft. Silicon and Synapse, a developer that would later change its name to Blizzard Entertainment, was developing a Super Nintendo version of Justice League Task Force. Condor ended up pitching the idea for *Diablo* to Blizzard, and halfway through the resulting development process Blizzard's parent company acquired Condor, renaming us Blizzard North. Throughout a tangled history of corporate juggling and ownership changes, Blizzard North has remained a very independent group. Our staff has grown steadily from about 12 at the start of *Diablo* to 24 at the start of *Diablo II*, and finally to our current group of more than 40. We concentrate 100 percent of our efforts on game development. To help keep this focus, Blizzard's headquarters in Irvine manages other functions, such as quality assurance, marketing, public relations, technical and customer support, as well as the operation of the Battle.net servers. Our parent company, Havas Interactive, deals with business functions such as sales, manufacturing, and accounting.



Much time was spent perfecting Act I since it would likely be used in a beta test or demo. The Amazon was the first character to be completed.

1. *Diablo II* is still *Diablo*. A constant theme in previews and reviews of *Diablo II* was that we didn't change anything; it was more of the same. At first that struck us as odd. We kept less than one percent of the code and art from the first game. We rewrote the graphics engine, changed all the character classes and skills, shifted and expanded the setting, reworked and added to the magic items, brought back only a handful of our favorite monsters, and designed a ton of new gameplay elements, such as running, hirelings, left-click skills, and random unique monsters. Why, then, did everyone think it was the same thing? In the end, we decided just to take it as a compliment. The play-testers and reviewers meant they were having exactly the same kind of fun that they had in the original game.

Both *Diablo* and *Diablo II* provide a constant source of simple pleasures, many of which are perhaps too basic and obvious to mention in evaluations and reviews, but which are fundamental to their success. We used the term "kill/reward" to describe our basic gameplay. Players continually kill monsters and get rewarded with treasure and experience. But the rewards don't stop there. We offer a steady stream of goals and accomplishments to entice the player to keep playing. There's always a quest that is almost finished, a waypoint almost reached, an experience level almost achieved, and a dungeon nearly cleared out. On a smaller scale, we tried to make every single action fun. Moving around inventory items produces pleasing sounds. Monsters die in spectacular fashion, like piñatas exploding in a shower of goodies. We

strove for overkill in this sense, in that players are constantly on the verge of something great - only a few mouse-clicks away from a dozen interesting things.

Diablo II retained *Diablo*'s randomly generated levels, monsters, and treasure. This obviously allows for better replay potential, but also serves to make each player's game his or her own. Players feel an ownership of their own game experience in that they are actively generating a unique story. It's enjoyable to tell friends about what you have just done in the game, knowing for sure that they have not done the same thing. Simply following an online walk-through won't help them accomplish goals without effort.

Finally, *Diablo* and *Diablo II* are easy to play. We used what we call the "Mom test": could Mom figure this out without reading a manual? If we see new players struggling with how to sell items, we look at how they're trying to do it and make that way work too. We strove to make the interface as transparent as possible. You want to open a door? Left-click on it. Want to move to a target location? Left-click on it. Want to attack a monster, pick up an item, or talk to a non-player character? Well, you get the idea. It's amazing how many games have different controls and key combination for all these actions when simpler is always better.



The architecture in *Diablo* combines aspects of many different cultures in order to arrive at an interesting mix that doesn't look too much like any single one. Here, the buildings of Travical from Act III are based on Mayan and Aztec references.

2. Blizzard's development process. Blizzard's development process is designed to ensure that we make a great game. While our goal is to meet the milestones we set, our process, in terms of design and business, is structured to allow us to wait until the game is as good as it can be before we ship it. We recognize that not all developers have this same opportunity, but many of the methods we use along the way are applicable to any development environment.

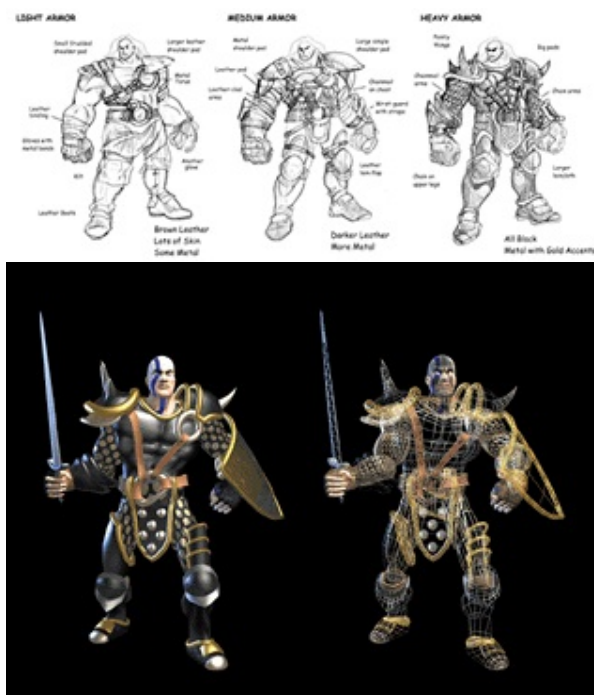
First, we make the game playable as soon as possible in the development process. Our initial priority was to get a guy moving around on the screen and hacking monsters. This is what players would be doing most of the time, and it had to be fun. We were constantly able to hone the controls, pathfinding, and feedback mechanisms during the entire length of the game's development. Most importantly, it allowed us to determine what was fun to do, so we could provide more of it, and discover what was awkward or boring, so we could modify or remove it. For instance, it became obvious very early that players would be killing large amounts of the same monsters, and those monsters would predominantly be attacking the players. This gave us the opportunity to plan for multiple death sound effects and additional attacking animations for each monster. If we hadn't experienced the core gameplay as early as we did, combat would have ended up feeling much more repetitive.

Also, we constantly reevaluate gameplay and features. Up until the very end, if we can make the game better we will, even if it means redoing big tasks. For instance, we decided that we didn't like the Bone Helmet graphics for the characters more than a year after having rendered them, but we went ahead and remade them, even though it took a couple of weeks and the collaboration of four artists. Only weeks away from scheduled beta testing, we scrapped our Act IV level layout schemes because they were just a bit too empty and similar. The last-minute fixes turned these levels into some of the best, befitting their climactic function. *Diablo II* took more than 40 people and over three years, essentially because we made two or three games and pared them down to the best one.

Another gigantic reason for our success is our open development process. We strive to hire people who love games, and we make games that we want to play. Every member of the team has input into all aspects of the game. Discussions around the halls and at lunch become the big ideas that shape the game. A programmer suggested to a designer the concept of gem-socketed, upgradeable weapons, which turned out to be a huge crowd-pleaser. A musician's dislike for the old frog-demon's animation inspired us to redo it. As a team, we don't have to wonder what our audience wants, because we are our audience. If we like the game we are making - especially if, after two years of playing it, we are not bored to death - the game is clearly going to be a winner.

3. Character skill tree. Our most revolutionary new idea was the character skill tree. For a character to attain more powerful skills, he or she

must master prerequisite skills. The ability for characters to branch into different areas of the skill tree, and to choose a level of specialization in each skill along the way, provides truly unique characters.



TOP: The player characters have modular armor of three varieties, light, medium, and heavy, which were mixed and matched to provide more individualized character appearances. "Paper dolls" created on paper and in Photoshop allowed mixing and matching of different pieces of armor to see how they worked together on the Barbarian.

BOTTOM: The Barbarian, translated from the sketches into a full, high-polygon model. Each part of a character's armor (the head, the torso, the legs, each arm, a weapon, and a shield) was rendered separately with in-house tools.

At the start of development, we planned to use the model from the original *Diablo*: characters would find and read books to learn spells and skills. Unlike *Diablo*, which had 28 spells shared by all three characters, we wanted to create a separate group of 16 skills for each of our five new character classes. This would definitely have been an improvement, but every character of a given class would still end up knowing all the same skills as other members of their class. Another problem was that players would likely be finding spell books for other character classes much more often than for their own. The skill tree solved these problems. The general idea was taken from the tech trees many strategy games employ. In strategy games, players advance by researching new technologies, which in turn open up further avenues of research. We adapted this to have our characters advance by choosing a new skill or strengthening an old skill every time they gain an experience level. Characters can generalize by choosing a wide variety of skills, or specialize by allocating many skill choices into a small group of skills. We also created a strategy element of choosing skills you might not use, just so you can get to one further up the tree later.

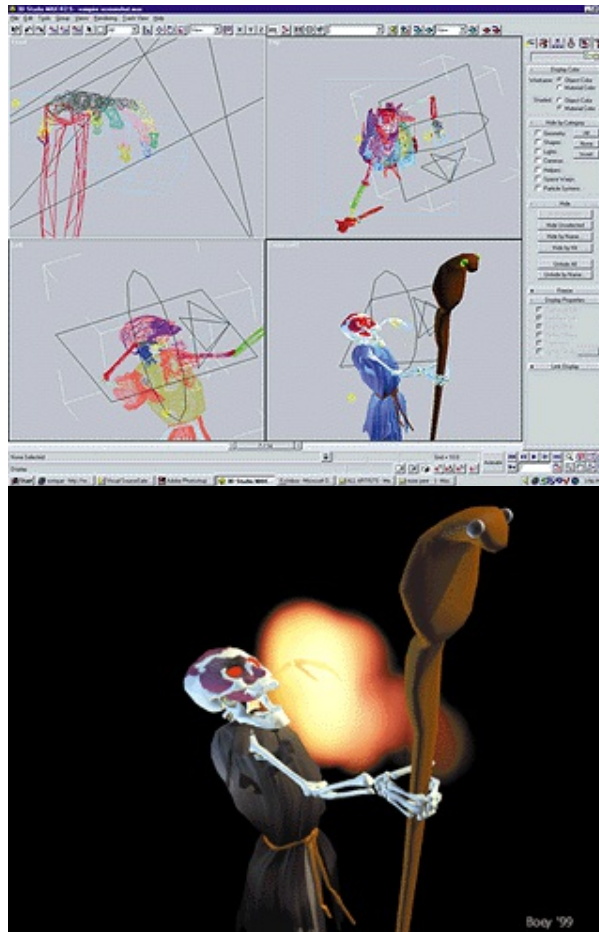
The end result of the skill tree is that one player can develop a Necromancer who kills monsters with a powerful poison dagger skill augmented by curses that cause monsters to fight each other, while his friend's Necromancer will summon hordes of skeletons to fight for him, and doesn't use any curses at all. The longevity of *Diablo II* will be enhanced by the endless strategies that can be debated and experimented with.

4. Quality assurance. The task of testing a game of *Diablo II*'s scope, with its huge degree of randomness and its nearly infinite character skill and equipment paths, required a Herculean effort. We found we could not play-balance the climactic fight against *Diablo* without actually playing the entire game up to that point, because we could not predict what kinds of equipment a character might have, or what path through the skill tree he or she may have followed. This meant 20 or 30 hours of play for all the different characters, with a good variety of skill sets and equipment for each. Whenever we changed the game's treasure spawn rate or experience curve, we had to test it all again. Further complicating matters were multiplayer and difficulty-mode balance. Would a party of five Paladins, each using a different defensive aura, be untouchable? After more than 100 hours of play, is a fire-based Sorceress unable to continue in "Hell mode"?

The QA team created a web-based bug-reporting database through which we categorized and tracked all bugs, balance issues, and gameplay suggestions. In the end, this list delineated more than 8,300 issues and suggestions. Well-organized teams of testers concentrated

on different aspects of the game, divided into groups that would specifically test character skills, item functionality, monster types, and spawn rates, or explore the countless variations found in the random level generation system. The members of the QA team became very good players and astute observers of the progress of the game. Everything worked much more smoothly than our experiences with the original *Diablo*.

5. Simultaneous worldwide release. In the past, Blizzard's strategy for shipping its game has been to get games on North American retailers' shelves as quickly as possible after the English version of the game went gold. With the original *Diablo*, we created our gold master on December 26, and some stores had it on the shelves by the 30th. Since *Diablo* was released, the percentage of international customers had increased substantially, and with *Diablo II*, we fully expect more than half of our sales to come from outside North America. With such a large number of customers located outside the United States, for *Diablo II* we decided that there would be significant advantages to coordinating the U.S. release to coincide with the rest of the world, not only to build anticipation for the product, but for the benefit and satisfaction of our customers as well.



Characters and monsters, such as this Vampire, were created in 3D Studio Max. An in-house tool would render the files from many different angles (eight for all monsters, 16 for player characters), and export them in the file formats used in the game.

If we release a game in the United States first, customers in the rest of the world don't want to wait a few months while we translate and localize it for their country. Due in part to the international climate fostered by the Internet, players around the world all know about the game at the same time and want to get it while it's hot. They might buy the U.S. version under the table or search out a pirated copy. Worse, they might lose interest by the time we release a localized version. *Diablo II*'s simultaneous worldwide release also allowed our marketing and PR departments to focus their efforts toward creating a frenzy of interest for the first week of sales. Although the simultaneous release was a

logistical headache, it was all worth it in light of *Diablo II*'s superb success.

1. Developing the new Battle.net. We have always been very proud that our company launched Battle.net with the original *Diablo*. Just a couple of months after *Diablo* shipped, Battle.net was the largest online game service in the world. At *Diablo II*'s launch, Battle.net had more than 6 million unique active users.

Despite the original *Diablo*'s success online, we knew as we began development that to create the type of multiplayer experience that we wanted to achieve in *Diablo II*, we would need to fundamentally change the game network. And, as we expected, this became one of our biggest challenges during development. We had to reinvent Battle.net's structure by melding existing technology with new programming and feature sets. This had implications across the board. We had to rethink everything - programming, hardware, bandwidth, staffing, online support, and how we could financially support this model while keeping it free.

Although the original Battle.net had been further modified to support Starcraft as a chat and matchmaking service, for *Diablo II* we needed much more: game servers where the Realm games would actually be played, secure character-data servers, and game tracking systems. Trying to shoehorn these elements in the existing Battle.net system proved very difficult. For instance, we planned to have character names represent players in Battle.net, but it was designed to handle chatting between account names. It took a lot of design and implementation time to arrive at our final system, where users see character names but have to send remote messages to account names.

We initially believed that working with the existing Battle.net would save us time, but in retrospect, we learned that melding technologies is a difficult process, and in some cases, recoding instead of integrating is the better course of action.

2. Launching the new Battle.net. The success of Battle.net after *Diablo*'s launch created a new challenge for us. When *Diablo* was released, Battle.net was a new online service. Basically, we were able to ramp up as more customers joined the service. When *Diablo II* shipped, Battle.net had millions of users. The level of anticipation was higher than for any of our other games. We were well aware of the expectations, and we knew that no other company had ever attempted to create and sustain an online service that could support the type of usage *Diablo II* would experience right out of the chute.

We spent countless hours preparing for Battle.net's *Diablo II* debut. We teamed with the best ISPs in the world, and conducted months of internal and external beta testing. We ramped up bandwidth and hardware. We beefed up the Battle.net, quality assurance, and support service teams.

Although we had more than 100,000 people testing the *Diablo II* Realms, having more than one million customers in just three weeks proved to be very different from beta testing. The beta test was very successful in uncovering many stability issues that were addressed before the launch. After the game shipped, we faced bugs that only appeared at much higher usage rates. The issues that we faced at launch were ones that could not have been simulated in a beta test of 100,000 people. It took a much larger influx of players to trigger certain situations.

Knowing the massive scope of what we were trying to achieve with Battle.net, we had measures in place at launch to help us deal with issues that arose as usage increased. For instance, we maintained both a team of programmers and the entire quality assurance department to solve problems as they appeared, and had our support team working overtime. We also had an action plan in place to increase hardware and bandwidth as needed.

In some respects, we are victims of our own success. We underestimated sales, but we also underestimated the allure of playing on the Battle.net Realms. By solving the cheating problem in *Diablo* and enhancing Battle.net with new features - such as the ability to see everyone's characters in the chat room - we seem to have attracted a larger share of Battle.net players than with any of our previous titles.

3. Graphics. Shortly before *Diablo II* shipped, we began noticing some feedback from customers about the resolution of the graphics in the game. They were frequently labeled "outdated" or "pixellated." The shame is that the technology choices we made eclipsed the recognition of the fantastic job the artists did. We put a lot of effort into creating characters, monsters, and landscapes with a lot of unique character. The game displays an incredible amount of action happening on-screen in an easy-to-follow manner. Still, with all the negative reaction, we probably should have done it differently.

When we began producing art for *Diablo II* in mid-1997, we investigated a lot of options. We mocked up a 3D engine and checked out voxel systems. It didn't take us long to go back to what we used in *Diablo*: 2D graphics at 640x480 resolution with 8-bit color depth. At that time it was still the only way to get eight characters, upwards of 30 monsters, and upwards of 100 missiles all interacting on the screen at one time without sacrificing detail and atmosphere.

The graphics criticism caught us by surprise. We thought (and still think) that the game looked great.



Some of the many locales in Act II:
The Sand-Maggot lair (top),
Jerhyn's Palace (middle), and the

We probably should have built in a scaling technology to take advantage of hardware that could display the same graphics at higher resolutions. In any case, *Diablo II* will probably be our last 2D game.

4. Tools. We developed the original *Diablo* with almost no proprietary tools at all. We cut out all the background tiles by hand and used commercial software to process the character art. Spells and monsters were balanced by verbal estimates ("Hey, lets make the lightning about ten percent weaker."). *Diablo II's* vastly increased scale required much better tools, and we made some, but not enough.

In many cases we created tools to speed up content creation, but then abandoned them. Too often, we decided to keep using inferior tools because we thought we were almost done with the game, and didn't want to take the time to improve them. Unfortunately, in most of these cases, we were not really almost done with the game, and in retrospect a couple of weeks' worth of work would have helped in the year or more of development remaining.

The greatest deficiency of our tools was that they did not operate within our game engine. We could not preview how monsters would look in the environments they would inhabit. We couldn't even watch them move around until a programmer took the time to implement an AI. Even after that, an artist would have to hassle someone to get a current working build of the game to see his creation in action. Our sound effects engineers ended up painstakingly creating .AVI movie versions of animations in order to synch sounds with actions. Our lack of tools created long turnaround times, where artists would end up having to re-animate monsters or make missing background tiles months after the initial work was completed.

We should have made tools that let us create content within the game engine. Instead of just handing off a set of animations and hoping they looked all right when dropped into the game, artists should have had the ability to position and orchestrate their creations themselves. The extra tool development time would have been more than offset by increased efficiency and higher-quality work.

5. Save-game methodology. As much as we tried to make a frustration-free game, we seem to have failed some people with our save-game scheme. Eschewing the common save-game feature we used in the original *Diablo's* single-player mode, where every facet of the game state can be saved to files and reloaded at will, we opted to make all modes behave more like *Diablo's* multiplayer game. In *Diablo II*, we do not save the world state. Reloading the game resets the location of monsters and treasures every time. The character is placed in the town he or she last visited, not in the wilderness or a dungeon.

Although this choice was slightly controversial around the office, it had a lot of advantages. For one, players could not get stuck, unable to progress further. At any time you can restart in town, refight the same monsters for more experience and loot, and return to a difficult area when ready. We created a waypoint system that allows characters in new games to return quickly somewhere close to where they quit the previous game. Finding new waypoints is a rewarding mini-goal during play. We also wanted to discourage the type of play where players feel they must always save the game right before a difficult section, then constantly die and reload until they get lucky and make it through. Finally, it was just easier to make single-player games and multiplayer games work the same way, and multiplayer requires the method we used.



Monsters have 14 possible classifications of animation, from basics such as Walk, Attack 1, and Death, to the seldom-used Block, Run, and four Special modes, reserved for miscellaneous animations. *Diablo* is the only monster who uses every animation category available.

Sewers (bottom). Background elements were created and rendered in 3D Studio Max. The rendered files were cut into tiles and assembled into modular "rooms" with an in-house tile-editing tool. The game engine reassembles the rooms to provide a randomized game environment.

A lot of players don't like our decision. They feel it is too inconvenient to have to fight their way back through the same areas and monsters. Many also want the opportunity to experiment with skill choices and equipment purchases, then later revert the game back to an earlier state

if they don't like the results. There are good points on both sides, and we probably didn't spend enough time developing alternatives.

The Final Word

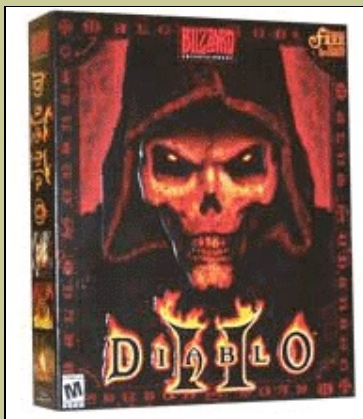
Many more things "went right" than could fit in that section. Our internally controversial plan to tell a separate but parallel story through our cinematic sequences seems to have succeeded, and the workmanship and quality of these sequences has set a new standard. Our marketing and PR departments did a fantastic job building customer awareness and creating a frenzy of interest. *Diablo II*'s music is outstanding, and along with an amazing array of sound effects, contributes hugely to the atmosphere of the game.

The development of *Diablo II* is a remarkable success story. We got the opportunity to make the game we wanted to make - and the game we wanted to play. *Diablo II* turned out to be a great game, one that many of us still play every day. Initial sales figures are phenomenal, and reviews have tended to be better than those of its predecessor. We have gained a lot of experience that should help us make even better games in the future.

The only major downside to *Diablo II*'s development was the inhuman amount of work it required. A yearlong crunch period puts a huge burden on people's relationships and quality of life. Our biggest challenge for the future is figuring out how to keep making giant games like *Diablo II* without burning out. As a start, we are hoping our experience will help us do a better job scheduling and managing the workload. We also believe that taking the time to make better tools will make things easier at the end of projects.

Although I tried to avoid personalizing this article, I am extraordinarily proud of the entire development team. *Diablo II* could not have happened without all the superb individual efforts, the incredible creativity, and the whole team's dedication to the project, for which they have earned my gratitude, and no doubt that of the legions of players who enjoy the game.

Game Data



Diablo II

Blizzard Entertainment

Publisher: Blizzard Entertainment

Full-Time Developers: 40

Length of Development: 3 years.

Release Date: June 28, 2000.

Platforms: PC and Macintosh.

Hardware Used: Typical programmer workstation: 500 MHz Pentium II running Windows NT with 128MB RAM and 9GB hard drive. Typical artist workstation: dual 500 MHz Pentium IIs running Windows NT with 256MB RAM and 14GB hard drive.

Software Used: 3D Studio Max, Photoshop, Microsoft Developer Studio/Visual Studio and SourceSafe

Notable Technologies: Glide, Direct3D, RAD Game Tools' Bink, DirectSound3D, and Creative Labs' EAX.



[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved