Game Development and Production

Erik Bethke

Wordware Publishing, Inc.

Library of Congress Cataloging-in-Publication Data

Bethke, Erik.
Game development and production / by Erik Bethke.
p. cm.
ISBN 1-55622-951-8
1. Computer games--Design. 2. Computer games--Programming.
3. Project management. I. Title.
QA76.76.C672 B47 2002
794.8'1526--dc21 2002153470
CIP

2320 Los Rios Boulevard
Plano, Texas 75074

Printed in the United States of America

ISBN 1-55622-951-8

10 9 8 7 6 5 4 3 2 1
0301

All inquiries for volume purchases of this book should be addressed to Wordware Publishing, Inc., at the above address. Telephone inquiries may be made by calling:

(972) 423-0090

Chapter 5

What Is a Game Made Of?

The Extended Development Team

Before you tear off into preproduction of your game, I want to show you all the parts that go into a game. Whether your background is art, programming, design, marketing, or sales, you will tend to view a game project as a medium of art, software with game design, a game design in motion, or a product to be marketed or sold. The big picture of game development involves a team effort of many individuals spanning dozens of professions all across our industry and spilling into other industries. When you see what it takes to make a modern commercial game, I hope you get a more balanced view of the various roles to be played to carry out a hit game.

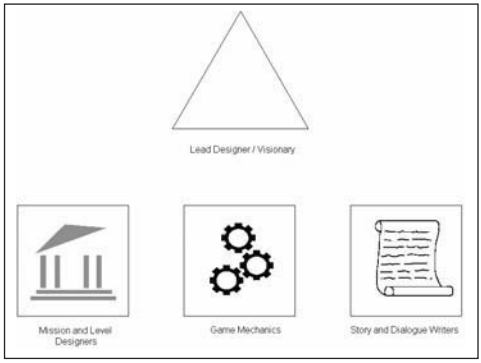
That Lever 2000 soap commercial is bouncing around my head right now with its silly jingle of all your 2,000 parts. So, following that jingle, let's take a look at all of the parts of a game.

Game Production Parts

Surely a game project is all about producing a great game. If not for the developers, there would be no product to sell! I am biased as I am a developer, and so yes, I do think game development is the most critical component of a successful electronic entertainment product. However, the developers hold a sacred trust given to them by the rest of the project stakeholders that they will be able to develop a compelling and competitive game, on budget and on time. This is a sacred trust that has been violated more times than it has been honored. We developers must perform to the best of our ability to deliver the strongest game on time and on budget.

Design Parts

1. Lead Designers/Visionary
2. Game Mechanics
3. Level/Mission Designers
4. Story and Dialogue Writers



The flavors of game designers

Where Do Lead Designers Come From?

We have to design a game first and foremost. Some games have a key visionary who has been kicking around an idea for a long time; others are more of a collaborative process with a leader. There is probably no single more difficult task in the industry than being able to create an original game of your own design and see it through to commercial release (only a nitpicker would point out that seeing your game become a mega-hit would be harder). Each game has its own story of how it got to be funded and made. However, it is usually the publisher or the studio head of a successful game development company that has finally arranged for all the business points to be in place in order to kick off their game.

If the publisher suggests the game concept, then the developer will supply the lead designer. Often the founder of a game development company will act as a lead designer on the project.

The lead designer's job is to coordinate the design staff in the effort to create timely, thorough, compelling game design specifications that the rest of the team can readily use and is readily understood by the game's publisher and other key stakeholders. The lead designer is not responsible for designing the whole game; rather it is the lead designer's role to be a director and sculpt not only what goes into a game, but also what does not belong and should be cut. (In practice, the lead designer also picks up any design tasks that the rest of the team is not able to do.)

How Do You Nail Down the Game Mechanics?

Each game usually has a lead game mechanics designer. This person often has a game programming background, as programmers are the ones most likely to implement the game mechanics in the code. This person receives direction from the lead designer, solicits engineering feasibility from the programming staff, and confers with the mission or level designers to find out their requirements. Depending on the type of game, the game mechanics designer often plays with Excel, trying to achieve a rough balance to the game and simulating portions of the game to get an idea of how some of their mechanics will play both for single player and multiplayer.

Who Are the Level and Mission Designers?

Some games have levels, others have missions, and quite a few have neither. Whatever game you have, it can almost always be broken down into a series of smaller challenges, puzzles, levels, or missions for the player to complete. Level and mission designers are sometimes programmers writing scripting code for a mission. Sometimes these

designers are artists laying out tiles of a map and designing triggers, and sometimes they work in pure text, describing to others how the game should be laid out.

Story and Dialogue Writers Are Writers for Interactivity

Writing a compelling narrative that is formatted for the high degree of interactivity found in games is a wholly different skill than writing the narrative of a short story or novel or a motion picture screenplay. A writer for games needs to spend a lot of time with the lead designer for direction on where to take the story arc, and he or she needs to spend even more time with the mission and story writers to determine what is possible and not possible to do in the scripting language, map editor, or level building tool.

Writing natural sounding language for characters is not the same as just listening to people talk and writing it down; it is a talent for having an ear that sets the right rhythm of tone and balance for their characters to speak in a fantasy world in a believable manner.

I am discussing design roles that people will play, not saying that each project will literally divide its design tasks into discrete people; in other words, designers will cross over back and forth through these roles.

Coding Parts

I detailed game designers first, as the designers define the spirit of the game; however, I have often been caught saying the ultimate designers on a project are the programmers and the artists. The designers can write documents and create specifications until they turn blue, but the game will not be anything other than what the programmers and artists create. I am not trying to cast programmers as an uncooperative bunch; I am a programmer myself. What I am trying to say is that the programmers and artists are very special people and often need to be convinced of the designer's vision. Most often the final implementation is a blend of the designers', programmers', and artists' collective vision.

The programmers' roles are to obviously create the code: the 3D engine, the networking library, art asset converter, and such, to realize the vision for the game. Games are often late, over budget, or buggy as I mentioned earlier. Games are hardly ever late two months while they wait for the tile artist to get her act together, and games are hardly ever late by a month because the audio guys have not mastered your sounds yet. It is a rare project that is delayed due to sheer asset production deficiencies, and even when that occurs the programmers are not idle. Why? Because electronic games boil down to just code—code with art, code with sound, code with gameplay, yes, but it is still just code. Even with code being the main deliverable, why does it always have to be late? This is an issue that is larger than the game industry. In Steve McConnell's *Rapid Development*, he writes that 50 to 90 percent of general software engineering projects are significantly late. Software engineering projects, in general, are chronically failing. The reason for this is that we game developers are part of a larger industry—software development—that is in turn an immature branch of the engineering discipline. The processes in specifying software, the processes for

creating software, and the processes for testing software and even establishing skill levels in programmers have yet to be established! You have to be a licensed engineer to pilot a ship for commercial transport, to build a bridge or a skyscraper, or even analyze the soil on a hill for a single-family dwelling. In fact, in California and in most states you must have a license to cut someone's hair. No one needs a license to write code.

The idea of licensing game programmers may seem, at first, ridiculously out of place in the game industry. The lifeblood, the very soul of the industry is founded on clever people dropping out of whatever they were doing before and putting their heart and soul into creating a fun game. Why do I advocate the clearly un-fun part of getting a license to write code?

Imagine a future of game development where each game project has a licensed software engineer as the lead programmer or technical director (with the license administered much like a professional engineering license). With this type of person a very important safety structure has been put into place. Someone is responsible for the technical soundness of a project, and not only is her name and reputation on the line for this project, but her license to operate as a professional engineer could be revoked if she is shown to be manifestly negligent in her role as a technical director. I know I am way out on my own here with this opinion, but I really think this would protect not just the programming staff from unreasonable schedules, but the *publishers* themselves. They could lay down some outline of a feature set, quality level, budget, and timeline and say go make the game, but it would be so much stronger if they had to have the signature of the lead programmer (a licensed software engineer) to sign off on the project before the project could continue past preproduction and into production.

Microsoft employs a version of this method where Microsoft employees have to sign off on a developer for technical, artistic, design, and project management competence before any funding of the team can commence.

Well, enough of my diatribe on the merits of licensing programmers, let's go see what they actually do on a project.

Lead Programmers and Technical Directors

The lead programmer has traditionally been the most experienced programmer on the team (from the 1970s through the 1980s, he or she could have been the only programmer). The lead programmer usually takes on the programming tasks that are the most challenging of the project. The quintessential examples of lead programmers are John Carmack of id and Tim Sweeney of Epic. These guys are usually the heroes of the projects, and many teams are structured around the lead programmer.

Some games tend to have a large programming staff, such as the massively multiplayer game Ultima Online or EverQuest, or the single-player/multiplayer game Neverwinter Nights with *over 25 programmers*. These large projects typically employ a technical director that oversees the programmers and reports directly to the project manager. The technical director title implies much less coding being

performed by the individual and more management of programmers and code creation. Sometimes smaller projects employ a technical director when the lead programmer is handling a tricky part of the project she does not care for or has no time for, or is otherwise not suitable for project management. Another model is to have a series of "assistant leads" who are all responsible for different aspects of a programming task—i.e., functional leads—who each in turn manage a few programmers and who ultimately report to the lead programmer. This is the model at BioWare and at Taldren.

The lead programmer is like the queen in chess; she might be your most productive programmer on the project, but you must use her time wisely. Technical directors, on the other hand, act as scouts on behalf of the programming staff, looking ahead, lining up dependencies between programmers, and coordinating the development of the software.

The rest of the programming positions I describe below are not necessarily distinct humans on every project; rather they are common programming roles that most projects have. A lot of projects, for example, have the 3D graphics programmer and the lead programmer be one and the same, or the game mechanics and user interface programmer could be the same person.

Game Mechanics Programmer

The game mechanics programmer is the one who converts the "real meat and potatoes" of the game design into playable code. This person usually models the physics of the game world, how objects such as weapons and potions work, and how the protagonists and antagonists function.

The game mechanics programmer can usually be seen near one of the project's designers, debating the merits of the designer's weapon mechanics and such. The game mechanics position is usually a mid-level programming job that ambitious scripters and mission programmers often grow into.

The great thing about being the game mechanics programmer is you are the one who really puts the game into the game. You are the first one to see a lightning bolt strike the ogre, the first to see a tank shell a building, and the first to see the health pack heal the character. This is a fun job.

3D Graphics Programmer

The 3D graphics programmer is one of the most highly respected positions in the industry. 3D graphics programmers must have a strong comfort level in mathematics including calculus, vector and matrix math, trigonometry, and algebra. The 3D graphics programmers enjoy seeing their work come vividly into being in lush 3D graphics, immersing the player in environments they can only dream about.

Artificial Intelligence Programmer

The demands on the artificial intelligence programmer vary from game to game and from genre to genre. Steven Polge, now working with Epic, has written some truly impressive bits of AI code such as the Reaper bot. Also, the AI programmers are usually the folks who have the proper skills to write scripting languages and other tools used by the designers.

User Interface Programmer

The user interface programmer is the person who has the tricky job of developing the software that bridges the game mechanics of the fantasy world with a slick implementation of the user interface through the controls, in-game panels, and HUD elements, as well as the shell or navigational menus. The UI programmer is the expert on the UI library and usually maintains it by extending its functionality. The UI programmer position is one that is likely to have been gained through experience in the industry. UI programming is often hard to get precisely right and is often underappreciated.

Audio Programmer

The audio programmer is the person who codes up the 3D sound effects, the voice-over tag system, and the music playback system. Often this position is a light position due to strong, widely used audio libraries available such as the Miles Sound System from RAD Tools.

Tools Programmer

Michael Abrash once told me that id spends greater than 50 percent of its programming resources creating tools. This is a significant statement. Most game companies do not commit this level of programming resources to their games. BioWare has a large tools department as well, over ten people, who make tools for all of BioWare's games. They have found this saves a lot of time and rework by designers and artists. The fact that id is arguably the most successful developer ever, with many mega-hits of their own as well as a prosperous licensing program that includes other mega-hits such as Half-Life, seems to say that every programmer on the project should be a tools programmer half of the time.

Most teams do not have full-time tools programmers, although if the team is part of a larger house, there might be a tools department. Still, every solid game company builds up its own toolset over time to get graphics on the screen, get audio out the speakers, and get the characters in the game to have interesting behavior.

A game development organization should have short-term and long-term tools production goals. I suggest a Gantt chart produced in MS Project be printed out and hung on a wall to indicate the internal tools development in your organization. This visibility will help everyone see how the tools are integral to the growth of your team and how things are planned to get better in the future.

Mission/Level Editor Programmer

The mission editor programmer is just one of the tools positions; however, for many games with a mission or level editor, the editor will be released to the public with the game's release. Developing a mission editor or level editor that is robust and easy to use is the work of creating another piece of commercial software. The UnrealEd level editor for the creation of Unreal Tournament levels by Epic is a fine example of a 3D solid constructive geometry modeling and scripting tool that is extremely powerful, robust, and easy to use by both industry professionals and by fans who want to make new content for their favorite games. Some development houses organize a world-building tool as part of the main

game team, and others put this work in the tools group if they were rigorous in the technical design of the world editor to make it truly useful for other game projects.

Network, Server, or Client Programmer?

The network programmer writes the low-level and application-level code to get games running between a small number of players using modems, a local area network, or across the Internet. In the past the network programmer had to master a variety of protocols such as IPX, and serial and modem protocols. Modern games are now run almost exclusively on TCP/IP and UDP, the networking protocols of the Internet.

The multiplayer architecture of games can be broken down into two main structures: peer-to-peer and client-server.

Peer-to-peer structures have all of the player machines simulating their own copy of the game and use a variety of algorithms to keep the states on the different computers as close as possible. The peer-to-peer machines all talk directly to every other computer in the network. The bandwidth required to service this model of game grows exponentially with each added player. That is an unfortunate side effect as you try to handle more players.

The client-server structure divides up the computing of game simulation into a server, which handles the actual simulation, and the client, which is the viewer, or browser, of the world events. There are several benefits to this structure, including the fact that the bandwidth requirement grows only linearly with the number of players, and the game can also be protected from quite a few forms of cheating by having it run on a trusted and secure server. (Remember, in a peer-to-peer game each machine is running its own copy of the world and has authority on some portion of the world. This authority can easily be abused by running a rogue version in the peer-to-peer network.)

Why are not all games client-server? Arguably they all should be; however, depending on the game, the client-server architecture is much more complex and requires divorcing the simulation and the presentation along much stricter object-oriented lines. Today's massively multiplayer games are a prime example of the complexity of client-server games. Literally dozens of machines, running a score or more instances of servers, carry out different operations such as player authentication, version checking, cheat detection, game simulation, chat hosting, database transactions, and more. Peer-to-peer games are much more similar to traditional single-player games with the exception of the games periodically making corrections to be more in line with each other's view of the world.

Art Parts

The artists of an electronic game may wear a host of different titles just like the programmers. Games used to have a single artist drawing the character sprites and the world backdrops for these electronic heroes to carry out their missions. In the earliest days the programmer, designer, and artist were one and the same person. Starting in the mid-'80s small teams of artists, usually no more than three, would work on a project. Starting in the early '90s game projects grew substantially in

their art requirements and budgets. Famous examples of these are Wing Commander IV by Origin, where over $10 million was spent by Chris Roberts on chasing the dream of the fabled movie-in-a-game; Mario64, rumored to have a budget of over $20 million; and finally the Japanese epics in the Final Fantasy series and Shenmue, which have had gargantuan budgets.

Artists are now differentiated by their skill sets. It is interesting to know that many artists can build 3D models of the most arcane objects quite accurately and swiftly without being able to sketch them. The domain of the artist now covers a wide enough area that you will need to plan your art team carefully to be sure you have enough bandwidth of skill and talent across your art requirements.

Art Director

The art director is the manager for the art team, scouting ahead to be sure that project dependencies are taken care of ahead of time and that the artists produce their art assets on time for the rest of the game project. The other, arguably more important role is to look at every art asset as it is being constructed to be sure it is consistent in quality and theme with the rest of the game.

The art director job should be given to the artist with the most industry experience, tempered with people skills, and the person who best enjoys the entire team's respect.

Concept Artist

The concept artist is gathering visibility. In the past a few sketches would convey the look of the major characters and locations, and the game was off into production. Now with project budgets 10 and 20 times larger than in 1995, the stakes are much larger and the penalty for getting the art wrong is often fatal to a project. This is where the concept artist saves the day. High-quality black-and-white drawings are often colorized (color comp) to accurately convey to the art director, the producer, and the major project stakeholders what the look of an art asset will be before it is created. For example, on our Starfleet Command series, we needed to create a black-and-white sketch for each and every proposed ship model we wanted to introduce into our Star Trek game. These black-and-white sketches first made the rounds of the team to be sure we liked them, then the sketch went on to Interplay's upper management, then on to Paramount's interactive licensing director, and on to even Rick Berman, the producer of the *Star Trek* television show and movies now at Paramount. Only when we received approval from all these folks did we start to colorize the sketch and start the approval process once again for the colorized sketch. Once this was approved, we were permitted to actually begin work on an art asset that would make it into the game. (The resulting 3D model would of course need to make this same approval-seeking trip.)

This approval process is even more stringent at LucasArts on *Star Wars* properties, and Japanese games are very much oriented around the concept artist, such as Yoshitaka, best known in the game industry for his work on the Final Fantasy series.

2D Artist/Interface Designer

The 2D artist is an expert in classical sketching and painting. These artists are capable of painting backdrops, creating character portraits, and creating tiles and sprites for use in non-3D game engines. These artists used to use Deluxe Paint in the golden age of game development and have now moved on to Photoshop, Illustrator, and other packages.

Even in a 3D game, the 2D artist is an incredibly versatile and important member of the team, producing high-res artwork for ads and marketing, and helping to create assets for a promotional web site, install graphics, and countless more elements of 2D art.

The interface designer usually is an expert 2D artist with a strong sense of functional aesthetics. This artist will make just navigating your game's menus an exciting and fun activity. The interface designer is a key team member; be sure you have one, or don't make your game. Sometimes designers and programmers with strong visual design skills can successfully fill this role. This area of art is the most closely tied to your game—the game design, the game mechanics, and the look of the game. And these areas see the most change of any art asset. For these reasons, I strongly recommend against outsourcing your interface design art assets—get the best person you can and work with him full time.

3D Modeler

The 3D modeler was the highlight of the show around 1994-1997. At this time artists with experience in the industry were almost invariably 2D artists who were clever or stubborn enough to get their 2D visions articulated into a painfully small set of pixels using tools such as Deluxe Paint on the Amiga and later the PC. These artists on the whole were not prepared to handle the technical requirements of operating a 3D modeling package. Instead, a strange hybrid programmer-artist with a fascination for things 3D was required to operate the early arcane 3D packages. These artists were also in prime demand in the movie industry, and the scale of wages paid there made it very difficult for the game companies to recruit them over to games. In these years game projects had to train their 2D Deluxe Paint artists slowly to use early versions of LightWave and other technical 3D packages.

Over time the packages got much stronger and easier to use. College courses now teach 3D Studio Max, and in general people have had time to learn how to use the 3D modeling packages. 3D modelers are still highly respected members of any game team, but it is more balanced now with the other key art positions.

Character Modeler

The character modeler is a specialized breed of 3D modeler. Some strong 3D artists are competent at making mechanical things such as spaceships, tanks, and architecture, while others seem to lean towards the organics of characters. Low-poly character modelers have a special understanding of how the detail of the character will come to life in the texture stage to make the most economical use of their polygon budget.

Texture Artist

The texture artist, like the concept artist, is now a highly visible element of your art team. Games are almost always constructed out of polygons with textures on them. The sophistication of the modeling packages is so strong now, the texture phase of creating a 3D object is usually estimated at three to four times longer than the actual building of the model. The texture artist is a 2D artist who can "skin" an object in his mind and create a compelling set of textures to "paint" that skin on the 3D model.

Animator/Motion Capture Studio

Animation comes in two broadly different categories: character/animal/monster animation and everything else. Rotating antennas, windmills, and radar dishes are good examples of the everything else category. Animating a windmill is an almost trivial task for an artist on your team, while animating the snarl on a goblin's face is an entirely different task.

JARGON: *Key framing* is the technique of using a 3D modeling package to set key frames to have the engine interpolate between.

JARGON: *Motion capture* is using a special matrix camera to record the movements of a real human actor wearing a motion capture suit that has funny reflective balls attached to it. Most projects that use motion capture also use key framing for part of their animation duties.

To animate a character, two different solutions are at your disposal: key framing and motion capture. Key framing is the older, more established method of animating your characters. Key framing excels at animating cartoon characters and monsters and for extreme movements—motions that are impossible to capture with a human actor. Animating by key framing is an entirely different skill set from 3D modeling, texturing, or sketching. If your project will involve characters that need to be animated, be sure your team has enough competent animators to get the job done; animation can be a slow art.

Motion capture is the buzzword—this is the state of the art. Humans move with very subtle grace; studying a motion-captured movement will reveal how much the whole body moves during the walk or the swing of a bat. Motion capture's largest drawback would have to be cost in both dollars and time spent massaging the data into usable form. This field is constantly improving, and there are half a dozen competitors in the field. In Chapter 33 I will show you in depth what you need to know about motion capture including how to get a successful bid.

There is quite a bit of technical drudgery involved in smoothing out all of the details of the character's model and animations—dealing with the skeleton, motion capture data, prop bones, and a host of tiny, necessary details. Some studios divide this work between the modelers and the animators depending on the nature of the task, and other studios like BioWare have dedicated folks called character riggers who handle these types of tasks.

Storyboarder

If your game is to have any movies or cinematic sequences, it is important that your team have a storyboard artist. The storyboard artist will be able to design and articulate the scenes in a sequence for internal and external review before committing to costly live action or resource-intensive computer-generated sequences. Show the movies to the publisher, show them to the team, and work it all out ahead of time through simple boxes and captions. Most storyboarders are accomplished concept artists but not necessarily.

Audio Parts

Audio assets come in three main flavors: sound effects, music, and voice-over. In the beginning there were only crude sound effects performing buzzes, beeps, and whistles. We now have full Dolby 5.1 3D sound. Music has come a long way from clever timing of beeps to compositions by film composers performed by 50-piece live orchestras. And voice acting is now an art form performed by stars like Patrick Stewart and contracted under the authority of the Screen Actors Guild.

Voice-Overs

Voices in a game really bring it to life. Compelling voice acting reinforces every other element of interactivity by having the actors speak to your character. The tutorials for Starfleet Command went from being a dry introduction to our gameplay to being the most compelling *Star Trek* moment I ever experienced with George Takei performing Admiral Sulu teaching me to command a starship. I remember when Origin's Strike Commander was released for $50, but an additional speech pack was available for $20 more. That is a testament to wacky product strategies as well as a testament to the compelling depth voice adds to a game.

The only way to get good voice work done is to work with an experienced voice-over director. A good director will know immediately where to secure the talent, the studio time, and the engineer, and get you the post-processed audio in a format you need. In Chapter 29 I will guide you through the process of getting high-quality voice into your game. The pleasant surprise of voice work is that it is probably the coolest element you can add to your game for the money, and it is essential in many role-playing games, which are dialog and VO intensive.

Sound Effects

Sound effect engineers are wizards at listening to one sound and finding clever ways to stretch it, compress it, twist it, and come up with precisely the sound you need. Sometimes they will Foley—that is, record your sound effect from the actual object generating the sound. Sometimes the sound engineer will record some other sound and then twist it around just for your game.

Sound effects are an excellent target for outsourcing as only the larger developers with three or more concurrent projects can keep a sound effects crew productively working. Chapter 30 contains an interview with a sound engineer so you can see what it will take to get strong sounds into your game.

Music

Some games spend a lot of effort on music, and it really gets the emotional hooks into the player when the music is first-rate. Music is probably the most popular and oldest art form worldwide. Nearly any emotion can be invoked with compelling music. There are two options: synthesized music and music that is performed live. We spent nearly $100,000 on the score and 30-piece orchestra performance for Starfleet Command 2. The music was very special; all of the sounds are richer and fuller bodied when performed by humans versus a synthesized chip. That being said, a single musician can create extremely strong music with a professional synthesizer and software. Chapter 28 will discuss outsourcing music in detail and give you plenty of leads to be sure your game has the emotional impact of high-quality music.

Management Parts

Management of a game project is the most critical component in my experience. In recent private email with other studio heads in the industry, the consensus was that a developer is limited in number of teams not by programmers or artists, but by quality producers/project managers. That being said, the management of a game project is often shared by a group of individuals with different responsibility sets.

Line Producer

The line producer coordinates countless small tasks that one by one are not very challenging, but taken as a whole is a daunting amount of work that needs to get done every day. If a project lacks a line producer, the efficacy of every team member will be compromised by a little distraction at a time. The line producer will often supply the team with food when the hours are forced and late; will get design documents printed and sent overnight; and will often coordinate getting builds out to the publisher and to beta testers. The line is a critical function that should be filled by a line producer, instead of your art director on Mondays, your 3D graphics programmer on Tuesdays, and so on.

JARGON: *Builds* and *revs* refer to interim functional versions of the game distributed for testing to internal and external testers.

Associate Producer

The associate producer is found on larger projects in a single team company, and all ompanies with multiple teams need an associate producer. Publishers also structure themselves with an executive producer managing a group of titles and an associate producer on each title performing day-to-day management. The associate producers have an interesting combination of a lot of responsibility and little authority. The associate producer is the understudy of the executive producer. The business negotiations, contracting, and human resource decisions will be carried out by the executive producer, but in almost every other aspect of the game project the associate producer will have a strong contribution to make. The associate producers are often burdened with the dreary task of updating the schedule and reporting on task tracking. The associate also helps communication between all team members and is usually the strongest advocate

for the game. In truth, each studio has its own name for the hierarchy of managers in the organization such as assistant producers, senior group producers, and project planners.

Studio Head/Executive Producer

The studio head at a game developer and the executive producer at the publisher each have the same fundamental job on a game project: be responsible for planning and executing the project in a profit-producing manner.

Studio heads are almost always the founders of their own companies, those who have risen through the ranks and are industry veterans and who have paid their dues and made money for their publishers in the past. In the case of Valve, Gabe Newell brought lots of project management experience from 13 years of creating software such as Microsoft Windows. Studio heads run small companies—game development shops—and have to simultaneously be game designers who are passionate about their games, software managers who respect technology, and businessmen who are savvy enough to get a good publishing deal. Some developers such as id and Epic have divided the role of the studio head into a more practical split of one person running the business and another acting as the project leader for the game.

The business development executive at the publisher often supplies the executive producer on the publishing side with a game project and game developer lead. The executive producer's job is to then complete the evaluation of the developer and project to determine its suitability for production. If the executive producer is confident the project should go forward, he will negotiate the key terms with the developer and work to help the project meet its first internal green-light or assessment milestone. If the project passes, then the executive producer's job is to oversee the project's progress through the reports generated by the associate producers and by looking over builds of the project in progress. The executive producer is often called upon to maintain the relationship with any licenses and is sometimes involved in contracting external vendors. The executive producer is the person most visible inside the publishing company for the game's success, while the press and the fans tend to focus on the game developer.

Producer

As a game development studio grows into two teams or larger, the role of the producer becomes critical to the effective execution of the studio's projects. The producer is the person who will manage the project at a larger development studio, allowing the studio head/executive producer to concentrate on strategic company issues.

Quality Assurance Parts

Quality assurance (QA) is another critical component of game development. The single best way to test your game is of course to play it and play it until it is solid and as much fun as you know how to make it. The problem with this method is that it will take a very long time for a single person to play through a game in its entirety (which may not even be possible), and a single person will make errors and have a bias.

The industry has yet to come up with a unified testing method that is known as the best practice employed widely. Instead each developer and publisher and indeed each game project tends to have its own QA process. Microsoft appears to be the organization that exerts the most effort in a rigorous QA process.

Most small developers do not have a full-time QA staff, as they would only see useful work roughly half of a project's lifetime. Larger, multiteam development companies can often gainfully employ a full-time QA staff. For example, BioWare employs a full-time QA department of over ten people, which supplements the even larger QA teams at their publishers, reducing the errors sent to the publishers to speed up development and saving the developers themselves from having to test their own stuff, instead allowing them to focus on finishing new content/features. Smaller developers often cross-train the line producer and associate producer to be the first line of QA with a backup of team-wide testing days.

Publisher QA Parts

All high-profile commercial games receive a considerable amount of professional testing by the publisher's internal QA department. This department follows the guidelines set by the publisher's management and works as efficiently as possible to report defects in content and quality to the developer prior to commercial release. Most commercially released games have anticipated release dates that are difficult to postpone in the case of a late project or a particularly buggy one. These internal QA teams are trained to report the severity of the defect and generally create high-quality bug reports that have items prioritized for the development team's attention.

QA Lead

The QA lead is the person who leads the efforts of the QA staff. The QA lead is always a former tester who showed promise of superior skill in organization and communication. The QA lead coordinates getting new builds or revs of the game in progress.

The QA lead also proofreads all reported defects from her team and discards duplicate and erroneous reports and often rejects reports back to the reporting tester, requesting clarification and/or testing. The QA lead is almost always an aspiring game designer or producer and often includes extensive commentary on the game's content in order to gain visibility for possible promotion. This is because most publishers have an outdated, poor concept

of the QA staff and treat them as low-skilled, low-paid workers, leaving those workers with little choice for a career in QA. Instead they are actively trying to strike out into development or some other role in the game industry. A notable exception to this is Microsoft, which seeks out folks with college degrees and pays well for its QA positions.

Main Team

The main QA team is the team that will monitor the game's progress from the time the game is initially submitted to QA through release and often into post-release. The main QA team will go through stages of varying productivity in direct relation to the development team's ability to respond to the bug reports in a timely fashion. This team is generally referred to as the QA team even though there are many other potential testers of the game. The main QA team will often rotate in fresh team members as a natural process of other games finishing and employee turnover.

Multiplayer Team

Games with significant multiplayer gameplay often have a QA team dedicated to testing this functionality. This is more common with PC games, as console games tend to have much more limited multiplayer gameplay. The multiplayer team is used to be sure all of the modem, LAN, Internet, and matching options are thoroughly tested. Bugs associated with multiplayer code are often more difficult to track down and report; this allows testing of the single-player campaigns and missions to continue on in parallel. In the same manner, individual members of both the multiplayer team and main team are specialists in a particular portion of the game such as a chapter or character class or playable race.

Fresh Teams

The problem with having dedicated main teams and multiplayer teams who look at the same game from three months to a year is that their ability to discern fundamental problems with gameplay and usability are compromised fairly quickly as they learn the game and lose the critical insight of a new player. It is still important to have efficient teams who know what the game is and what the last reported set of bugs were so they can quickly turn around a bug report to the development team. However, fresh teams are often introduced to a game the closer it comes to shipping, depending on QA resources available internally to the publisher.

Compatibility Team

The compatibility team is often a dedicated team of QA members who happily rebuild computers all day while testing the major functionality of your game. These guys have very little work to do on a console! The compatibility team usually has a standardized checklist of hardware and operating systems the publisher considers commercially important to support.

Localization Team

Also, all big games are localized into various markets, and native speakers of these languages will be employed to QA both the accuracy and the quality of the localization of the game.

Beta Testing

Beta testing is testing performed by unpaid volunteer fans who want a first peek at an upcoming title and who are excited by the opportunity to improve a game before its release. At first many publishers were apprehensive that a beta version of the game would become widely pirated and steal sales from the release version of the game. Or in the case of weaker titles, many publishers consider it a shrewd strategy to avoid the beta testing stage. Perhaps the most successful beta testing programs are run by id; examples of these are Doom and Quake Test. These first-person shooters had multiplayer gameplay and no single-player missions. Even with only three or so maps to play test, these "tests" by id produced more hours of fun and gameplay than most games ever achieve in their final release. I personally played several hundred hours of Quake Test before Quake was released—sniff—thank you, id!

Bottom line, if you want to make a great game, run a beta test and fix your game until beta testing proves you are ready for release. In recent years the advent of the massively multiplayer game has required extensive beta testing. These massively multiplayer games require hundreds if not thousands of concurrent players to analyze how the server will respond to the stresses of full release. These thousands of beta testers are also required to smooth out the authentication, account management, and game balance to avoid having paying subscribers complete the beta testing period. The sheer costs of these games and the limited rigor employed to date on beta testing programs still results in the pressure to release these games to the public and endure two to six months of painful post-release beta testing that strains the faith of your hardcore, early-adopting fans.

Beta Testers

The beta testers are almost always the fans who showed up on your message board when you first opened up shop. They often have beta testing experience or have heard about beta programs and will sometimes be quite proactive in their effort to secure a seat in your beta testing program. The number one rule with beta testers is to communicate with them; failure to do so only creates an expectation in the beta tester's heart that they are part of the development team, only to find out that their voices are unheard.

Beta Testing Program Manager

To facilitate this communication with the beta testers, one of the development team members—often the associate or line producer—takes on the role of beta testing program manager. This is a very stressful job. The time period that beta testing takes place is during the final months of a project when everything must come together. The beta testers are anxious to see their reported defects fixed in the very next version of the game and are quite vocal about new features they want and how they want the game to be balanced. In Chapter 23 I will discuss the mechanisms and techniques the beta testing program director should employ for a successful beta test.

Business Parts

Making games is big business. Depending on how you look at the numbers, the console game market (hardware and software) along with the PC game market generates more revenue than the box-office receipts of all of Hollywood's films annually.

There are a lot of different business executives who are involved in a game project; here I will present the major roles.

Business Development Parts

Business Development Executive

The business development executive is casually called the "biz dev."

> **JARGON**: "Biz dev" is the short name given to the business development executive at a publishing company.

When developers go around pitching games to publishers, they first need to get the approval of the publisher's business development executive before the game is sent to a green-light committee.

The biz dev person keeps a close eye on what is going on in the industry and is the first to know about games in development that are looking for a publishing deal. The biz dev person often negotiates the key terms of a game publishing contract.

Publisher CEO and President

A chief executive is responsible for all aspects of the game publishing corporation. Very often this individual has ten to twenty years in the game industry and has a well-developed instinct for making great games (not infallible though). Making sure that your game is visible and impressive to this key executive at green-light meetings ensures the highest level of support the organization can bring to bear for your game.

Studio Heads

Founders, lead programmers, visionaries, game makers, CEOs, presidents, head coaches—whatever you call them, studio heads are the chief decision makers at a game development house. Studio heads generally have five to fifteen years of experience in the game industry and at least one hit title under their belt where they held a strong lead role. Studio heads most commonly come from programming and design backgrounds, although there are some medical doctors of considerable renown running BioWare. Artists are the majority shareowner at id, and Gabe Newell of Valve had an extensive background of software development at Microsoft.

Studio heads decide the fundamental structure and working environment for their studios based on past experience. The studio head is intimately involved when a game project is starting up and is usually the salesperson pitching the game to the publishers. Studio heads are generally the most qualified team leaders in their organization and spend a lot of their time training new producers to run teams and subteams.

Lawyers

Both the publisher and the developer need the best lawyers they can afford. Each contract is unique, and while a

publisher's contract is the fruit of many painful relationships, the developer should be patient and exercise great care in negotiating terms. This is something you do not want to try on your own.

WARNING: Do not negotiate a publishing contract without the aid of a lawyer who has strong experience in electronic entertainment publishing contracts.

Lawyers are actually good people who help you understand clearly what a contract is and is not saying. Understanding what you are agreeing to before you sign a contract is a fundamental safety mechanism for both the developer and the publisher. In Chapter 27 I provide a list of law firms who are used by different studios.

Licensing Parts

Many games are based on licenses such as comic books, novels, movies, and sports stars. In turn, games themselves are licensed to create strategy guides, action figures, T-shirts, and movies. Publishers may have their biz dev executive manage the licensing of a game, or they may have a full-time staff member for routine licenses such as strategy guides.

Promoting, Buying, and Selling Parts

Sales? Is that not the job of the teenage clerk at the local Electronics Boutique? Well, yes of course, but well before a gamer walks into a computer game store, a sales force has made the larger sale of the game to the buyer agent of the retailer.

The decision on the retail buyer's part of how many units of the game title to order on release depends on how hot the title appears to be, the wholesale price, and the influence of any number of incentive programs that have been negotiated between the publisher's sales force and the retailer's buying force well before the game's release.

Sales Executive

Each publisher has a top executive in charge of sales. This person has a lot of influence on the ultimate sales of a game. The executive in charge of sales has a budget that goes by several different euphemistic phrases such as "marketing development funds"; this budget is spent to buy shelf space at retail. This is a pretty strange concept to people who are unfamiliar with the industry—that the publisher not only needs to absorb the risk of funding the development of the game and its packaging and marketing, but also must completely absolve the retailer from any risk. Selling games is a consignment business.

The retailer will put the product up on the shelves, and if it does not sell quickly enough, the retailer simply sends the product back and gets its money back. Retailers take maximum advantage of this relationship when a highly anticipated game is released by ordering as many units as the publisher will deliver. It sounds great when you have an order of 200,000 units from CompUSA for your game, but if your game fails to meet expectations, CompUSA will not hesitate a moment to send 160,000 units back to you—all marked up with their price tags—and simply order more later. Those 40,000 units you sold at CompUSA effectively had the packaging and shipping costs of 200,000 units, which wipes out much of

the margin from those 40,000 units that did sell.

A careful study of some publishers' financial reports to the SEC will show periodic "write-offs" and "one-time charges." There can be a whole variety of reasons why a business is forced to report a loss on their books, but in the case of game publishers it is often massive quantities of returned games that they have accumulated for as many quarters as they could get away with. It is not unheard of to see six to ten quarters of accumulated returned product discharged as a write-off. Keep in mind that during those six to ten quarters this product was accounted for as revenue. This practice is not sustainable, and the stronger publishers do not do this. A strong sales executive should work closely with the publisher's chief financial officer to manage what is called "sell-in" to the retailers with the goal of having the highest "sell-through" to "sell-in" ratio.

JARGON: *Sell-in* is the number of units the retailers order or buy.
Sell-through is the good stuff; this is the measure of how many units of your game were sold through to consumers—a true sale.

Sales Force and Retail Purchasing Agents

Under the direction of the sales executive, the publisher's sales staff meets periodically with the retail purchasing agents, each of whom represents a different retail chain. Prior to calling on the buying agents, a publisher will often host an internal sales meeting to communicate their product's selling points to the sales force. These meetings can sometimes be fairly lavish with, for example, large ice sculptures and Klingon impersonators to get the sales staff pumped up and primed to handle the buying agents.

Press Relations Manager

The press relations manager will oversee how the game is communicated to the press. For large titles, this is a nearly full-time job, and a quality PR manager should be split across as few titles as possible—one to three titles at most. The PR manager will field all press inquiries, as well as inquiries by those claiming to be press. The PR manager will strategize and plan how the details of your game will be released to the press.

JARGON: *Buzz*—what the press, fans, and industry are saying about a particular title.

If PR has a solid date on when the game will ship, then PR can create a solid plan for ramping up the buzz in a steady, ever-increasing volume to peak just as the title is released. Releasing too many of your game's goodies too early will provide you with little to say later in the project, and interest in your title will sputter and fade before it is released. On the other hand, if you do not release enough information on your game to grab press and fan attention, it may be difficult to maintain the support of the executives at the publisher and other project stakeholders.

Trade Shows

The Electronic Entertainment Expo, or E3, is the largest show in North America for publishers to get their products implanted in the agents' minds. E3 is a vast show with tens of thousands of attendees strolling through hundreds of displays ranging from mini amusement

parks from the likes of Nintendo to a folding desk and some business cards from discount CD duplicators. Thousands of products will be on display and scores of tricks are used to try to get your attention, from the obligatory booth babes to breath mints that are rolled out like cellophane. E3 is a cacophony of sound effects, lights, noise, and people. For all of this energy E3 is the largest news reporting event in the game industry and next to the retail buyers, the game press is the second most important contingent of VIPs to grace the floor. These folks have conspicuous press ribbons dangling from their badge so you know when not to speak candidly (handy).

Like anything competitive, the press at E3 is out to get more viewers and readers. The larger the market share, the more their business will grow. Years ago the press were trying to figure out how to arrange their time more efficiently for those precious three days of E3; they wanted to be sure they looked at every hot game. It would be a minor tragedy if a competing magazine or site were to report on a major title that you failed to see at the show. So the publishers and the press put together a schedule of viewings and demonstrations for all of the large press. That might sound innocent enough, but if you think about it for a moment, you will realize that all of the major press walks into the show with a schedule of titles filling all of the required genres and platforms prioritized in order of importance. Of course this journalist will still walk the floor, but it will be between appointments or at the end of the show. This makes it really tough for the little games trying to break out, as they are not even on the list to be seen.

The Internet game sites have another pressure—real-time reporting. To keep up, all of the major game sites need to have nearly live coverage of the show in an effort to bring the show to the fans and of course to gain more viewers. Real-time reporting is hard for several reasons, not the least of which is that you need to have something to say. Here again, publishers and the press will work together to give the press an E3 package a couple of weeks before the show. This package always contains the best screen shots, plenty of quotable material, and occasionally a playable preview build of the game. The better journalists look at this material as just more information; the less rigorous journalists (or those with very little time) have been known to lift the majority of the quotable material and publish that in lieu of an original opinion on the game.

Other Trade Shows and Events

E3 is important and dominant no doubt; however, it is hard to get your message across to the buyers, press, and fans when there are 3,000 other titles. Publishers have been creative about how to handle this problem; they hold their own shows in one form or another. For example, Activision hosts its own show in Europe between E3 and ECTS (the major show in Europe) to be sure awareness is implanted before and more effectively than the ECTS show.

Interplay hosted a very cool event for three of the Star Trek games (one of which was Starfleet Command) on the Paramount Studios lot. Press from around the world came to view three

games hosted by George Takei. The trailers for all three games were shown in the posh Paramount screening theater, and a fine lunch was served where the press mingled with the developers for an extended Q&A period after the press had a couple of hours to play the games. It was a relaxed but focused event that gave those three games ample time with the press.

The Marketing of a Game

As you can see there is a lot of sales and promoting of a game behind the curtains, but what about the ads—the traditional form of selling a product? Of course games have ads; take a look at your favorite game magazine and it seems like half of the pages are full-page ads. And the online sites have banners, navigational bar headings, and a myriad of advertising terms for the various bits of electronic click-mes.

Like the press relations manager, the marketing director for a game should not be spread too thin across many different games. The marketing manager will work with the producer and development team to craft the game's image in all of its various forms: print ads, banner ads, and the all-important box.

Just like press coverage, it is important not to create too much hype for the game and then fail to deliver on time. Publishers are getting much more savvy and are scheduling their marketing campaigns to kick off only when the ship date is known with confidence.

The marketing manager will also be responsible for getting your game shown at smaller venues such as the GenCon game convention held in Milwaukee. The marketing manager will coordinate strongly with the press manager and sometimes supervises the press activities. Sometimes it is considered a peer position, and in some places the same person is overloaded with both jobs. In particular, the marketing and press managers will be working closely with any playable demos that are to be released, making sure they are cover-mounted on the game magazine CDs and that the retail stores carry a supply of demos in a display.

Hardcore Fans

It is commonly known that hardcore fans and the word-of-mouth sales they generate is the largest factor in the number of games you will sell. Hardcore fans are eager to check up on the progress of their favorite game at the developer's web site, interact in the forums, and beta test. If they like the game, they can be responsible for not just the sale of the box they buy for themselves, but for the six or eight boxes that they have convinced their LAN party to play with. Or in the case of console games, the hardcore early adopters get the game first and invite people over to play. I have met fans who have sold ten, twenty, and more titles just for their passion of the game. Hardcore fans are always looking for the best in games; they also have a bunch of friends the industry calls casual gamers and mass-market gamers. These casual and mass-market gamers ask for recommendations from the hardcore gamers. The hardcore gamers will in turn recommend the titles they feel comfortable with. This is just common sense, but what it means is that Blizzard's Diablo was perfectly poised to capitalize on the

streamlined interpretation of the computer role-playing game genre where just light taps on the left mouse button looted catacombs and vanquished elemental evil. Valve's Half-Life laid a heavy story on top of the first-person shooter genre dominated by id (in fact they licensed id's Quake engine) to produce a mega-hit. And depending on how you measure it, Half-Life and its free, fan-created mods Counter-Strike and Day of Defeat are the most popular online games. These games are simply the most approachable, solid, and just plain fun games you can buy. If you want your game to sell, study how narrow the feature sets of Mario64, Half-Life, and Diablo really are, and how well and deep these few features are executed.

Manuals and Strategy Guides

Games need to have a manual, and if the game is considered a potential hit, then no doubt a strategy guide will be produced for the title.

Manual

How the manual gets written varies from publisher to publisher and from game to game. The most common method is to use an experienced contract manual writer. This person receives a copy of the game about four to six months before release and interacts with a member of the development team while writing to create the most accurate manual possible before a game ships. Another common method is for developers to create the manual given that they are the most familiar with the game's functionality. The biggest challenge in creating a manual is that rarely does one have the luxury of waiting until all features have been frozen and all stats in the game have been balanced. This results in almost all manuals being vague in some areas and fairly narrow in the scope of just providing use of the controls of the game, rather than how to play the game. Now enters the strategy guide.

Strategy Guide

The strategy guide fills a niche role in the game industry, providing detailed stats, walk-throughs, strategy, and tactics to complete a game. Writers of strategy guides have various stories, but it is not as simple a job as playing your favorite game and writing up all the nifty hints and secrets. What really happens is that the publisher of the game and the publisher of the guide work together to get builds of the game to the strategy guide author as early as practical in the project. Essentially, the guide author is a beta tester too; this makes the job of writing the definitive guide more challenging as the stats, missions, puzzles, and various parts of the game are still in flux. For instance, even the ultra-high-profile game Gran Turismo 3 (GT3) for the PS2 contains many discrepancies in the pricing of various upgrades between the U.S. version of the game and the U.S. strategy guide. GT3 shipped in Japan well ahead of the U.S. version and as such there was a little more time to produce an accurate guide. Despite this there were still discrepancies.

For our own Starfleet Command: Orion Pirates, the strategy guide writer of SFC2, Dennis Green, returned to write the most thorough guide possible. His project came under stress when we at Taldren overlooked some of his

requests for information during the final push. Unfortunately, after we were able to catch up and provide him with the information he needed, several strong chapters of the book had to be cut to reduce paper costs for the guide. It is a tough market to make money when work already created has to be cut.

Manufacturing Parts

I am astonished at how quickly a PC game can reach the store shelves. Do you know how fast a publisher can take the final gold master from the hands of the QA lead and deliver a shrink-wrapped retail box in an Electronics Boutique shop in the local mall? Five days. That is right, in five days a 30-cent recordable CD from the local OfficeMax can be turned into $70 million of merchandise on store shelves in the form of Diablo II. This is perhaps the quickest a game can reach the store shelves and usually only occurs at a fiscal quarter end for the publisher—most especially Q4 for the holiday shopping season!

To accomplish this a publisher has an operations manager who keeps his eyes peeled looking for the strongest vendors for CD duplication, manual printing, box printing, and assembly. This is quite a job, and normally they would like to see about 20 to 30 days to get the job done, so as to not have to pay for express drop shipments between the vendors. But when the end of the quarter is rearing its ugly face, the operations manager saves the day. Toward the two-thirds mark of your schedule, meet with the operations manager to nail down the firm dates for when they need everything—final box, final manual, and final posters and other goodies in the box. This is definitely an area of the project where it repays you in spades to be proactive and find out the due dates for these deliverables ahead of time.

Hardware Manufacturer Parts

Console Manufacturers

The console manufacturers assign a producer to oversee the development of each of the titles for the platform. The console manufacturer retains broad editorial approval rights for the game, and it is very important to follow their feedback to receive your ultimate approval for the gold master.

Hardware Representatives

Some of the coolest people to work with in the industry are the hardware vendors like SoundBlaster and NVIDIA. These folks are motivated to be sure that not only does your game work on their hardware but also that your game takes advantage of all of the features of their latest cards. What that means to a PC developer is a bunch of free hardware such as sound cards, video cards, joysticks, and speakers for use of the development team to test the hardware. These folks are best approached at their booths at the Game Developers Conference (www.gdconf.com). Tell them your story, where you are working, and what game you are working on, and if they feel that you are for real, you can get test hardware. Please do not abuse this if you are not making a commercial game and will not be making a genuine test of the hardware, as it will only make those resources harder to come by for the rest of us.

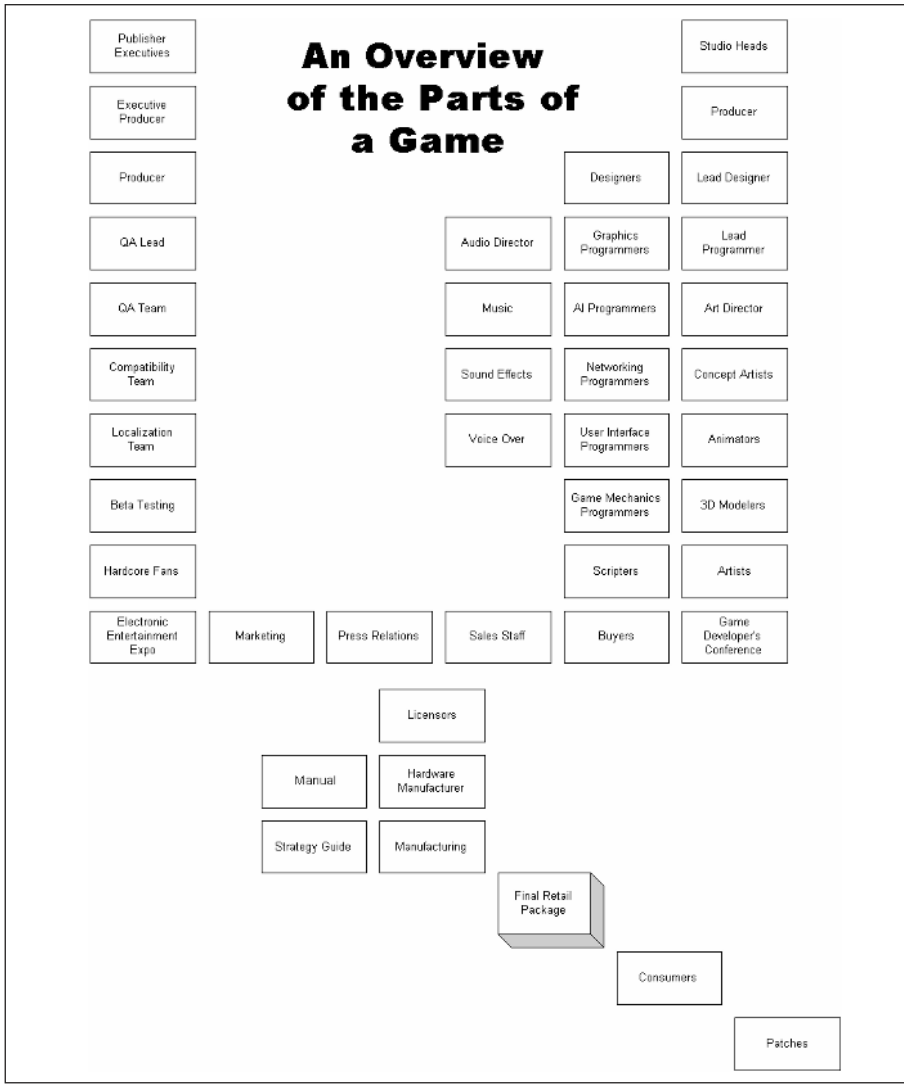
Post-Release Parts

Releasing a successful game to retail will be one of the most difficult things you accomplish in your professional career. After all of the cleverness it will take to get your project funded, staffed, and real; after all of the dedication to the craft during production; and after all of the blood, sweat, and tears it will take to drive a game through the final candidate cycle, you will find the day after you signed off on the gold master one of the most pure days in your career with no task that must be done now. Instead, you and the rest of the team will most likely disappear and rediscover what your family looks like and decide to talk with them—and sleep. After this much-needed rest is completed, is it time to dream up a new game? No, it is time for post-release.

Post-release involves patches, updates, answering questions on the forums, helping customer service field questions on the phone support lines, and combating cheating. For massively multiplayer games, these issues are much more serious as you are billing for a monthly service instead of a one-time purchase of a product. In fact massively multiplayer games have whole development teams called the "live teams" to maintain the software, add new content, act as gamemasters, and in general keep the product fresh and alive in the hands of gamers.

Having a bunch of fans is a very good thing; that is the whole reason for your work. However, a bunch of fans require a substantial amount of interaction and communication. At Taldren about six of the employees have taken the initiative to read our forums on a regular basis to field questions and moderate the forums.

Chapter 24 discusses the issues of post-release in detail with guidance from several studios on how to most effectively support the fans of your game.


An Overview of the Parts of a Game
Publisher Executives
Executive Producer
Producer
QA Lead
QA Team
Compatibility Team
Localization Team
Beta Testing
Hardcore Fans
Electronic Entertainment Expo
Marketing
Press Relations
Audio Director
Music
Sound Effects
Voice Over
Sales Staff
Studio Heads
Producer
Designers
Lead Designer
Graphics Programmers
Lead Programmer
AI Programmers
Art Director
Networking Programmers
Concept Artists
User Interface Programmers
Animators
Game Mechanics Programmers
3D Modelers
Scripters
Artists
Buyers
Game Developer's Conference
Licensors
Manual
Hardware Manufacturer
Strategy Guide
Manufacturing
Final Retail Package
Consumers
Patches