GAMASUTRA
The Art & Business of Making Games

PRINT

The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal

By Tim Ryan

The purpose of design documentation is to express the vision for the game, describe the contents, and present a plan for implementation. A design document is a bible from which the producer preaches the goal, through which the designers champion their ideas, and from which the artists and programmers get their instructions and express their expertise. Unfortunately, design documents are sometimes ignored or fall short of their purpose, failing the producers, designers, artists, or programmers in one way or another. This article will help you make sure that your design document meets the needs of the project and the team. It presents guidelines for creating the various parts of a design document. These guidelines will also serve to instill procedures in your development project for ensuring the timely completion of a quality game.

The intended audience is persons charged with writing or reviewing design documentation who are not new to game development but may be writing documents for the first time or are looking to improve them.

Design documents come in stages that follow the steps in the development process. In this first of a two-part series of articles, I'll describe the purpose of documentation and the benefits of guidelines and provide documentation guidelines for the first two steps in the process - writing a concept document and submitting a game proposal. In the next part, I'll provide guidelines for the functional specification, technical specification and level designs.

The Purpose of Documentation

In broad terms, the purpose of documentation is to communicate the vision in sufficient detail to implement it. It removes the awkwardness of programmers, designers and artists coming to the producers and designers and asking what they should be doing. It keeps them from programming or animating in a box, with no knowledge of how or if their work is applicable or integrates with the work of others. Thus it reduces wasted efforts and confusion.

Documentation means different things to different members of the team. To a producer, it's a bible from which he should preach. If the producer doesn't bless the design documents or make his team read them, then they are next to worthless. To a designer they are a way of fleshing out the producer's vision and providing specific details on how the game will function. The lead designer is the principle author of all the documentation with the exception of the technical specification, which is written by the senior programmer or technical director. To a programmer and artist, they are instructions for implementation; yet also a way to express their expertise in formalizing the design and list of art and coding tasks. Design documentation should be a team effort, because almost everyone on the team plays games and can make great contributions to the design.

Documentation does not remove the need for design meetings or electronic discussions. Getting people into a room or similarly getting everyone's opinion on an idea or a plan before it's fully documented is often a faster way of reaching a consensus on what's right for the game. Design documents merely express the consensus, flesh out the ideas, and eliminate the vagueness. They themselves are discussion pieces. Though they strive to cement ideas and plans, they are not carved in stone. By commenting on them and editing them, people can exchange ideas more clearly.

The Benefits of Guidelines

Adhering to specific guidelines will strongly benefit all of your projects. They eliminate the hype, increase clarity, ensure that certain procedures are followed, and make it easier to draft schedules and test plans.

Elimination of hype. Guidelines eliminate hype by forcing the designers to define the substantial elements of the game and scale back their ethereal, far-reaching pipe dreams to something doable.

Clarity and certainty. Guidelines promote clarity and certainty in the design process. They create uniformity, making documents easier to read. They also make documents easier to write, as the writers know what's expected of them.

Guidelines ensure that certain processes or procedures are followed in the development of the documentation - processes such as market research, a technical evaluation, and a deep and thorough exploration and dissemination of the vision.

Ease of drafting schedules and test plans. Design documents that follow specific guidelines are easy to translate to tasks on a schedule. The document lists the art and sound requirements for the artists and composers. It breaks up the story into distinct levels for the level designers and lists game objects that require data entry and scripting. It identifies the distinct program areas and procedures for the programmers. Lastly, it identifies game elements, features, and functions that the quality assurance team should add to its test plan.

Varying from the guidelines. The uniqueness of your project may dictate that you abandon certain guidelines and strictly adhere to others. A porting project is often a no-brainer and may not require any documentation beyond a technical specification if no changes to the design are involved. Sequels (such as *Wing Commander II, III,* and so on) and other known designs (such as Monopoly or poker) may not require a thorough explanation of the game mechanics, but may instead refer the readers to the existing games or design documents. Only the specifics of the particular implementation need to be documented.

Guidelines for the Game Concept

A game-concept document expresses the core idea of the game. It's a one- to two-page document that's necessarily brief and simple in order to encourage a flow of ideas. The target audience for the game concept is all those to whom you want to describe your game, but particularly those responsible for advancing the idea to the next step: a formal game proposal.

Typically, all concepts are presented to the director of product development (or executive producer) before they get outside of the product development department. The director will determine whether or not the idea has merit and will either toss it or dedicate some resources to developing the game proposal.

The director might like the concept but still request some changes. He or she may toss the concept around among the design staff and producers or open it up to the whole department or company. The concept can become considerably more compelling with the imagination and exuberance of a wide group of talented people.

A game concept should include the following features:

- Introduction
- Background (optional)
- Description
- Key features
- Genre
- Platform(s)
- Concept art (optional)

Introduction: The introduction to your game concept contains what are probably the most important words in the document - these words will sell the document to the reader. In one sentence, try to describe the game in an excited manner. Include the title, genre, direction, setting, edge, platform, and any other meaningful bits of information that cannot wait until the next sentence. The edge is what's going to set this game apart from the other games in the genre. For example:

"*Man or Machine* is a first-person shooter for the PC that uses the proven *Quake II* engine to thrust players into the role of an android space marine caught up in the epic saga of the interstellar techno-wars of the thirty-seventh century."

Breaking the introduction up into several sentences for the sake of clarity is acceptable. Just know that the longer your introduction, the more diluted your vision will seem.

Background (optional): The background section of your game concept simply expands upon other products, projects, licenses, or other properties that may be mentioned in the introduction; so it's optional. The background section is physically separated from the introduction so that readers can skip it if they already have the information presented. But the background section is important for licensed properties and sequels and concepts with strong influences from previously released titles in the same genre. If you intend to use an existing set of code or tools or to license a game engine, then describe these items and their success stories here.

Description: In a few paragraphs or a page, describe the game to the readers as if they are the players. Use the second-person perspective -- "you." Try to make this section an exciting narrative of the player's experience. Encompass all the key elements that define the core game play by describing exactly what the player does and sees. Avoid specifics such as mouse-clicks and keystrokes, but don't be too vague. You want the readers to become the player's character. Hover your detail level right above the GUI interaction. You would say something such as, "You scan your tactical radar and pick up two more bogies coming up the rear," instead of "You click on your tactical radar button and the window pops up revealing two bogies coming up the rear." The description section should make the content and entertainment value of the game obvious and convincing.

Key features: Your game concept's key features section is a bullet point list of items that will set this game apart from others and provide goals to which the subsequent documentation and implementation should aspire. It's a summary of the features alluded to in the description. These bullet points are what would appear on the back of the game box or on a sell sheet, but include some supporting details. For example:

"Advanced Artificial Intelligence (AI): *Man or Machine* will recreate and advance the challenging and realistic AI that made *Half-Life* game of the year."

Determining how many features to list is a delicate balancing act. Listing only one or two key features is a bad idea if you're doing anything more complex than a puzzle game; listing more than a page of features implies that the project would be a Herculean task and may scare off the bean counters. Listing too few features might sell your concept short; listing too many waters down the concepts' strongest features.

Keep in mind that you need not list features that are given, such as "great graphics" and "compelling music," unless you really think such features are going to be far superior to those of the competition. Great graphics, compelling music, and the like are the understood goals of every game project. On the other hand, if the particular flavor of graphics and music provides your game with an edge in the market, then you should spell that out.

Genre: In a few words, define the game genre and flavor. Use existing games' classifications from magazines and awards as a guide. For example, you could choose one of the following: sports, real-time strategy, first-person shooter, puzzle, racing simulation, adventure, role-playing game, flight simulation, racing shooter, god simulation, strategy, action-strategy, turn-based strategy, side-scrolling shooter, edutainment, or flight shooter. Then you can refine your game's niche genre with these or other words for flavor: modern, WWII, alternate reality, post-apocalyptic, futuristic, sci-fi, fantasy, medieval, ancient, space, cyberpunk, and so on.

Platform(s): In a few words, list the target platform(s). If you think the game concept is applicable to multiple

platforms, you should also indicate which platform is preferred or initial. If you intend multiplayer support on the Internet, indicate that as well.

Concept art (optional): A little bit of art helps sell the idea and puts the readers in the right frame of mind. Use art to convey unique or complex ideas. Screen mock-ups go a long way to express your vision. Art for the game concept may be beyond most employees' capabilities, so requiring it would limit the number of submissions; thus, it is optional. If a concept has merit, the art can come later from a skilled resource. Often art from previous projects or off of the Internet will jazz up a document. Just be careful with any copyrighted material.

Common Mistakes
Here are some common mistakes that developers make in creating a game concept:

- The concept is totally off base or inapplicable to the company's current plans. If you don't want to waste your time writing up concepts that get tossed, find out what the company in question is looking for and keep an ear to the ground for opportunities with which your idea may be a good fit.

- In terms of resources, the document asks for the moon. Try to keep your concept within the realm of possibility. Keeping the budgets down by suggesting existing tools or properties to reuse is helpful. Limit your ideas to that which can be accomplished in a timely fashion and with a reasonable budget. Limit experimental technologies to one area. Don't suggest revolutionary AI as well as a new, state-of-the-art 3D engine. If you are being solicited to produce the game concept, find out what the time frame and budget expectations are first.

- The document lacks content. Simply saying, "It's *Command & Conquer* meets *MechWarrior* where you order your 'Mechs in tactical combat," is insufficient. Your description has to explain the actions that the player will perform and make them seem fun. A good description might read, for example, "You order your 'Mech to fire at point-blank range on the exposed right torso of the Clan MadCat OmniMech." This kind of descriptive content will help mitigate misinterpretations of the core game play that you envision.

- The game isn't fun. A useful exercise is to break down all of the player verbs (such as shoot, command, run, purchase, build, and look) and envision how player performs each. Then, for each verb, ask yourself if it's fun. Then ask yourself if the target market would find it fun. Be objective. If the action that the player takes isn't fun, figure out another action for the player to take that is fun or drop the verb entirely.

- The game-concept document employs poor language and grammar. Don't make yourself look like an ass or an idiot. Check your grammar and spelling and avoid four-letter words and sexual innuendo. You don't know who will ultimately read your document or whom you might offend with some particularly expressive words. Even the macho, politically incorrect, culturally insensitive, slang-using manager with whom you exchange dirty jokes over a beer at lunchtime can get quite sensitive with documented verbiage.

- The designer gives up. Don't give up submitting ideas. You never know when one of them will take off. Persistence pays off, believe me.

Guidelines for the Game Proposal

A game proposal is a formal project proposal used to secure funding and resources for a game development project. As a game proposal takes time (and therefore, money) to do correctly, it should only be developed for promising game concepts.

The proposal is an expansion upon the game concept. Writing a proposal may involve gathering feedback and information from other departments, especially the marketing department (if it exists). You may need your marketing department to perform some market research and analysis on the concept. If the game requires licensing, you may need your finance and legal departments to investigate the availability and costs involved in securing the license.

The programming staff, typically senior programmers or the technical director, should perform an initial technical evaluation of the concept. They should comment on the technical feasibility of the concept and the programming areas that may require research. They should assess the risks and major tasks in the project and suggest solutions and alternatives. They should give a rough estimate as to the required research and development time and resources.

The game proposal should include a revised version of the game concept. Technical, marketing, and finance feedback to the concept document might force you to scale back the concept. It might also suggest modifying or adding features. These changes should not take anyone by surprise, as this is the first time that the concept has been subjected to major criticism and the collaborative process. Giving copies of the feedback and analysis to the director of development (or whoever asked for the game proposal) before they are folded into the game proposal or effect changes in the concept is a good idea. This process not only provides written confirmation that the concept has been reviewed by certain people or departments, but it arms the director with the knowledge to veto, alter, or otherwise approve any proposed changes.

The game proposal includes the following features:

- Revised game concept (preceding)
- Market analysis
- Technical analysis
- Legal analysis (if applicable)
- Cost and revenue projections
- Art

Market analysis: The marketing department and/or a market research firm, assuming your company can afford it, should compile this information. If you are compiling this information yourself, you should try to avoid pure guesses on numbers. Look for info on the Internet (www.gamestats.com is a good source) and use existing hits in the same genre as indicators for market performance.

- Target market: The target market is defined by the genre and the platform, issues that have been already addressed in the concept document. You can qualify this definition by mentioning specific titles that epitomize this market. The most successful of these titles will indicate the viability and size of the market. Also mention the typical age range, gender, and any other key characteristics. If this game involves a licensed property or is a sequel, describe the existing market.

- Top performers: List the top performers in the market. Express their sales numbers in terms of units, breaking out any notable data-disk numbers and any successful sequels. Include their ship date. You can be vague -- Q1 1998 or spring 1998. This research can go way back, so present your data in chronological order.

  List their platforms if they vary from the platform for the proposed game. However, because the markets change depending on the platform, you should always present some title of the same genre on the target platform, even if it didn't perform as well as the others. Such data may indicate a sluggishness for that particular genre of games on the platform. For example, turn-based strategy games may have great sales on the PC platform, but have terrible numbers on the Sony PlayStation. This list of top performers should indicate this discrepancy if you're doing a turn-based strategy game.

- Feature comparison: Break down the selling features of these top performers. Compare and contrast them to the key features described in the concept document. Try to provide some specifics. For example:

  Tactical Combat: In *Command & Conquer, Dark Reign,* and *Myth,* you order your units to attack specific targets and move to specific places or ranges for an advantage. Most units have a unique strength and weakness that become apparent during play, thus encouraging you to develop superior tactics. *Tanktics* has a wider variety of orders to allow you to apply superior tactics, such as capture, ram, and hit-and-run. Unit position and target selection become even more important due to terrain, movement, and range bonuses; firing arcs; and soft spots in rear- and side-hit locations. All of the units have distinct weaponry, armor, and speed to differentiate their strengths and weaknesses and encourage tactics. Not only do you learn to master these tactics over time, but you can also script these tactics into custom orders.

Technical analysis: The technical analysis should be written by a seasoned programmer, preferably the technical director or a lead programmer, and then edited and compiled into the proposal. Reviewers of this proposal will use this technical analysis to help them make their decisions. Be honest; it will save you a lot of grief in the end. Overall, this analysis should make the reviewers optimistic about the game's chance of succeeding.

- Experimental features: Identify the features in the design that seem experimental in nature, such as untried or unproven technologies, techniques, perspectives, or other unique ideas. Do not include features that have been proven by existing games, even if they are new to the development team. For example, if the team has never developed a 3D engine, don't list it as experimental. Rather, list it in one or more of the other categories in the technical analysis section. On the other hand, if your development team is working on a 3D engine using the theoretical system of quads, then this effort should be listed as experimental. Of course, by the time you read this article, quads could be in common use.

  Include an estimate of the time that it will take to bring the experimental feature to an evaluation state, as well as an overall time estimate for completing the feature. Experimental areas generally need more time in the schedule, so the more experimental features you list, the longer the schedule will be. While some companies shy away from such 18- to 24-month projects, many see these experiments as worthwhile investments in creating leading-edge titles. So tell it like it is, but don't forget to tell them what they will get out of it. Make them feel comfortable that the experiments will work out well.

- Major development tasks: In a paragraph or a few bullet points, make clear the major development tasks. Use language that non-technical people can understand. Major means months of development. Give a time estimate that assumes that you have all of the resources that you'll need to accomplish the task. You could also give an estimate of the resources that you'll need. For example:

  "Artificial Intelligence Script Parser: Three to four months with two programmers. The parser reads and compiles the AI scripts into lower-level logic and instructions that are executed at run-time."

- Risks: List any technical risks. If you don't foresee any technical risks, by all means say so. Risks are any aspect of research and development that will cause a major set back (weeks or months) if they fail. List technologies that, though they've been proven to work by competitors, your company has never developed or with which your company has little experience. List, for example, real-time strategy if your team has never developed a real-time strategy game before; or 3D rendering if this is your first foray into 3D. List any of the major development tasks mentioned previously if you perceive any risk. All untried off-the-shelf solutions (3D engines, editors, code libraries and APIs, drivers, and so on) should be listed as risks because they may end up not fulfilling your particular needs. Any development done by an outside contractor should also be listed, as that's always a big risk. When assessing risks, you should also indicate the likely impact that fixing or replacing the technology will have on the schedule. Indicate the time in weeks or months that the ship date will slip. List the time impact on specific resources. List any new resources (people, software, hardware, and so on) that would be required to fix it. This section may seem pessimistic, but it creates a comfort level for your document's reviewers - they will come away with the impression that the game implementation is under control, especially if they can perceive these risks themselves. Plus you'll have the opportunity to say, "You can't say I didn't warn you."

- Alternatives (if any): Alternatives are suggestions for working around some of these experimental or risky features and major development tasks. By presenting alternatives, you give the reviewers options and let them make the choices. List anything that might cost more money or time than desired but might have better results, or vice versa (it may cost less money and time but it may have less desirable results). Whatever you do, be sure to spell out the pros and cons.

- Estimated resources: List the estimated resources: employees, contractors, software, hardware, and so on. Use

generic, industry-standard titles for people outside of the company: for instance, the publisher or investor who might read your document. List their time estimates in work months or weeks. Ignore actual costs (dollars), as that comes later.

- Estimated Schedule: The schedule is an overall duration of the development cycle followed by milestone estimates, starting with the earliest possible start date, then alpha, beta, and gold master.

Guidelines for the Game Proposal, **continued**

Legal analysis (if applicable): If this game involves copyrights, trademarks, licensing agreements, or other contracts that could incur some fees, litigation costs, acknowledgments, or restrictions, then list them here. Don't bother mentioning the necessity of copyrighting the game's title or logo, as these are par for the course and likely to change anyway.

Cost and revenue projections: The cost and revenue projections can be done in conjunction with the finance and purchasing departments. This data should give the reader a rough estimate of resource costs based on the technical analysis's estimated resources.

- Resource cost: Resource cost is based on the estimated resources within the technical analysis. Employee costs should be based on salaries and overhead, which the finance or payroll department should provide. You can list these as average by title or level. Any hardware or software that you purchase should be listed as well, even if it will ultimately be shared by other projects or folded into the overhead budget. Use a table or embedded spreadsheet as it is easier to read and edit. For example:

Employee	Cost Per Month	Work Months	Total
2D Artists	$4,000	35	$140,000
Lead Artist	7,000	14	98,000
Level Designers	3,000	35	105,000
			Total: 343,000

Hardware/Software	Price	Qty.	Total
Graphics Workstations (PIII 500MHz/256MB/9GB/Voodoo2)	$4,200	3	$12,600
3D Studio Max Extended Site License (5-user pack)	3,000	1	3,000
			Total: 15,600

- Additional costs (if any): This section is an assessment of additional costs incurred from licensing, contracting, out-source testing, and so on.
- Suggested Retail Price (SRP): You should recommend a target retail price before your game goes in the bargain bin - pray that it does not. The price should be based on the price of existing games and an assessment of the overall value being built into the product and the money being spent to develop and manufacture it. Of course, your distributors will likely push for a lower sticker price or work some deals to use your game in a promotion that will cut the price even further, but that will all be ironed out later. Keep in mind that the higher the sticker price, the lower your sales, especially in a competitive genre where there's not as much demand as supply.
- Revenue projection: The revenue projection should show pessimistic, expected, and optimistic sales figures using the costs that you've already outlined and the suggested retail price. Other factors, such as marketing dollars and company overhead, should be left out of the picture as these are subject to change; if a minimum marketing budget is known, however, then you should certainly factor it in. Often the revenue projection is best represented with a pie chart or a bar chart. Be sure to indicate with an additional wedge or bar the costs incurred from any of the risks described in the technical evaluation and show totals with and without the risk assessment.

Art: If your game concept did not include any art, then the game proposal certainly should. The art should be created by skilled artists. Dispose of or replace any of the art in the concept document that was not created by the artists. The art will set the tone for the game. Assume that the readers may only look at the art to evaluate the proposal, so be sure that it expresses the feel and purpose of the game. Include a number of character, unit, building, and weapon sketches, in both color and black and white. Action shots are great. Include a GUI mock-up if your game is a cockpit simulation. Be sure to have good cover art. Paste some of the art into the pages of the documents, as it helps get your points across and makes the documents look impressive.

Presentation: The whole proposal, including the revised concept document, should be printed on sturdy stock and bound in a fancy report binder along with copies of the art. This document is to be presented to business people - you know, those people who walk around your office wearing fine suits to contrast with the t-shirts and cut-offs that you normally wear. Don't forget that you're proposing a major investment, so make the proposal look good and dress well

if you're going to handle the presentation. Preparing a slide show in a program such as Microsoft Persuasion is helpful for pitching your key points and art.

Common Mistakes
Some common mistakes in preparing a game proposal include:

- The analysis is based on magic numbers. Try to back up your numbers by listing your sources or explaining how you came up with them. Watching a producer pull some numbers from thin air and throw them in the document is almost laughable.
- The proposal is boring. This is a selling document. Don't bore the readers. Give them facts, but make these facts exciting, concise, and convincing.
- The proposal fails to anticipate common-sense issues and concerns. You should find out what this proposal is up against -- other proposals, competitive products, financing concerns, cost and time expectations, game prejudices, and historical mishaps. Your proposal will stand a much better chance of being approved if it addresses these issues preemptively rather than getting besieged by them during the review process.
- The proposal writer is overly sensitive to criticism. My experience might be unique, but don't be surprised if the major promoter for your game proposal decides to play devil's advocate. Make sure your proposal is solid. Believe in it and remain confident, and you'll weather the criticism and make believers out of those reviewing your proposal.
- The proposal writer is inflexible to changes to the game. Often during the process of gathering research or receiving criticism from the review committee, some reasonable suggestions will arise for changing or scaling back the design. Game development is a collaborative process. Take the feedback and change the design, or the game could die right there. The key word here, as tattooed on the arm of my buddy who served in the Vietnam War, is *transmute*. He had to transmute to stay alive and keep going, and your game idea may have to do the same.

This concludes part one of a two-part series on the anatomy of a design document. In the next part, I'll taunt you with some more colorful metaphors interspersed with some useful guidelines for creating functional and technical specifications and level-design documents. I'll also give you an overview of the document review process and how it facilitates the game development cycle.

Tim Ryan is a freelance game developer and software consultant. He has produced and designed computer games since 1992. His recent work can be seen in the PC game *MechCommander: The First MechWarrior Game of Tactical Command*. Please send your feedback, questions, and business inquiries to MuseOfFireProductions@yahoo.com.

Return to the full version of this article with comments