Writing game design documentation is a fundamental skill for a game designer, but it can easily turn into a “tail wagging the dog” scenario as designers get carried away with detailed documents when they should start prototyping their game.

Generally, the main purpose of design documentation is to communicate the vision and essential elements of your design to the other developers responsible for helping to implement it. We’re working on a solo project for this course, so our documentation effort is a bit of an exercise, but demonstrating strong documentation and professional writing skills in a portfolio piece like a game design document is important for game designers looking to get hired.

In this section I’ll explain some common pitfalls of game design documentation, and offer some tips for writing great documents.

Why You Shouldn’t Get Carried Away With Design Documentation

Students of game design sooner or later discover the notion of design documents: the blueprints or even ‘bible’ of a game describing exactly how it should function. Because this sort of thing sounds so important to the act of game design, and the only real prerequisite is using a word processor, writing design documents seems to be a point of emphasis for many aspiring game designers.

I find this problematic for a few reasons. First, the written word, the exclusive domain of the design document, typically invites grand ideas without consideration of the cost of the features described in the document. Even when you try to limit your **scope**, without prior development experience or trying to prototype what you’re describing, it’s very likely to underestimate the cost of the features you’re documenting.

Second, people can get hung up writing complex, detailed design documents. Trying to document every last element of a game idea before setting forth with development may seem like a noble pursuit, but there are some flaws with this idea. At some point, time is more effectively spent actually building the game rather than writing about it. What if the game idea you have spent so much time detailing has some fundamental flaw that can only be uncovered through playtesting? For this reason I recommend prototyping before getting too deep in documentation. Often times playing a game prototype uncovers the most fun additions, new directions, and best iterations you can make—as opposed to just closing your eyes and thinking about it. The earlier that design changes are made, the cheaper they are to implement. Late in a project’s production, it is not uncommon for design changes to simply be off the table due to the expense and risk of implementing them. This means you must prototype and playtest your idea as early as possible to discover what you got wrong on paper, or simply overlooked.

As a result of the iteration process mentioned above, often times the details of a game design slowly shift (for the better) from their documented state. If you have chosen a “design document as bible” mantra, this means you must actively keep all of your documentation updated. In the throes of production, tweaks are made to games at a rapid pace. Keeping the documentation updated can become a consuming task in and of itself.

Additionally, the more attached you get to a document, the less open-minded you may become to changes. This becomes even more of a risk when only one designer writes the documentation before trying to convince a team to help them develop the game. Writing something down becomes a way to insulate and over-protect game ideas, but most of the time ideas should be malleable. Games require **iteration** to become the best they can. If you are unwilling to make a change because it’s not written that way in a document, your game will suffer and it’s unlikely others will enjoy collaborating with you.

Finally, the longer and more intensely detailed your documentation becomes, the less likely anyone other than you is to read it. TLDR affects all professional communication, and that includes design documents.

Design Document Formats

If you’re working on a large project, rather than creating a singular uber-document containing every detail of your game, I suggest creating a handful of shorter documents, adding new ones as necessary. Big documents are unwieldy to read and navigate. Keeping a table of contents updated and managing a single huge document can soon spiral into a job in and of itself, which becomes quite a nuisance when you have other important tasks to finish on a deadline. Instead, starting with a simple overview document or outline of the game itself, and then creating new documents with details for each of your game’s systems and sections (levels, etc.) is a sound approach. Similarly, creating a wiki with entries for each of these can also work well. The popular source control hosting website GitHub has built-in support for wiki documentation. But sometimes more powerful and complicated tools like wikis can add unnecessary complexity to reading and writing documentation, so don’t be afraid to use a simple solution when it works.

It’s good practice to work from the cloud, using Google Documents or a similar service. This makes it easy for multiple collaborators to simultaneously edit the same document, and makes sharing with the rest of your team easy. This also makes it easy to work from multiple computers, in case you have a desktop and a laptop, etc. Further, it insures against any computer mishaps that might befall you, whether it’s a ruined hard drive or a power outage before you had a chance to save.

Our project will be small enough to cover in a single document.

What to Document

First, it's a good idea to write a brief overview or summary. This is useful to help you initially define the project from a high level and to acquaint the unfamiliar with the project. Start with the basics:

- Title (or working title; don't get hung up on this!)
- What platforms are the game designed for? Mobile / touch screens? Gamepads?
- What's the genre of the game?
- What's the camera perspective?
- What / who is the player character?
- What's the object of the game?
- What are the controls?
- Example walkthrough of a gameplay section
    - Ever read a walkthrough guide for a game you're playing? They describe the setting of a section of the game, and list exactly what the player needs to do to progress. As a designer, writing something like this for a game that doesn't exist yet can help others understand the vision for the game.

Some of this info may make sense to write out in sentence format, others may work best as bullet points or headings.

Next, consider the main features of your game. How does player movement work? How does player health work? How does scoring work? How does the player progress through the game? What enemies or obstacles are there? What items or weapons are there? Are there any other special features in your game? Each of these make for good sections to add to your documentation.

Documentation Tips

Design documents are technical writing and professional communication. As such, their most important quality is clarity. Your number one goal when writing in a design document should be to effectively, clearly communicate to your audience (that is, other developers responsible for implementing your design). An ideal design document communicates the designer's ideas so clearly that the party responsible for implementing it understands exactly what to do without ever speaking to the designer. This isn't always possible, or worth it, so sometimes it's best to just write down the basics with an understanding that you'll need to have a conversation with the other developers involved in implementing the feature before they start working on it. Consider a perfect instruction manual, either for a board game or some furniture you're trying to build.

Ideally, it is short and sweet, and after reading it you understand exactly what to do. Here are some tips you can take to improve your design documents.

Story != Game Idea

Many new designers tend to confuse story ideas for games for ideas for games themselves. Games can have stories, but stories and games are not the same thing. Interactivity is the defining characteristic of games, and your documentation should start by defining the interactive elements and making intended the player experience as clear as possible.

Later you can add sections of your design document that relate to the plot of the game, beginning with a high level outline.

TLDR

“To Long, Didn’t Read” is a real thing. You should always keep it in the back of your mind when writing professional communication, whether sending quick work emails or authoring a design document. When I give my students reading assignments, I know that if it appears long, the odds of them reading it drop significantly.

A high word count is not only intimidating to readers, but it also dilutes your document, and makes it less likely that someone will easily notice something important. The odds of a detail getting overlooked go up with each new detail you add. For this reason, write as sparsely as possible and think carefully about what you choose to commit to documentation. Simplicity is tantamount to clarity. If you can rephrase or present something in a simpler way, do it. Remember the wisdom of Strunk and White: “Omit needless words,” and “Avoid passive voice!” Strunk and Write wrote *The Elements of Style*, a handbook for writing, popular in many US high schools. If you haven’t read it, I highly recommend seeking it out, as the simple style rules will certainly improve your writing.

There’s a quote attributed to Mark Twain, “If I had more time, I would have written you a shorter letter.” Distilling your writing into its clearest, simplest form takes effort, and is often requires multiple revisions.

Short Sentences

In a similar spirit of avoiding TLDR, avoid long, run-on sentences. The longer your sentences, the more confusing your writing becomes. And remember, our primary goals are simplicity and clarity! Consider the writing style of William Faulkner, notoriously difficult to read for its long, stream of consciousness-like, run-on sentences. This can make for great literature, but design documents should not aspire to be artsy. They should be crystal clear. For these reasons, I recommend avoiding fancy punctuation and simply breaking compound sentences into separate, individual, shorter sentences. Another consideration: the longer a sentence, the more

likely it is for something to go wrong. Each extra clause is a chance for a mistake or confusion. Simplicity is always best, and the fewer parts something has, the simpler it is.

Built-in Heading Styles

Use built-in heading styles as you write. This automates table of contents and document outlining and helps ensure consistency in your formatting. You can customize the heading styles and apply changes to them all simultaneously. Using headings will help you break up your document into more digestible chunks.

Bullet Points and Numbered Lists

Sometimes complete sentences aren't even necessary for effective and concise communication. To an extent, I am a sucker for formality, but bullet points are perfectly fine for listing important information concisely. Just be wary of creating a list that's too long, includes too many sublevels, or having a list go through a page break so that the bulleted items are no longer visible with their heading or introduction (not an issue with online documents). If a list starts to grow out of control, perhaps there is a better way to catalog it, such as in a spreadsheet, or further subdivided into multiple sections.

Use numbered lists when documenting sequences, or to make it easier to identify and refer back to a particular list item. If there are more than a handful of bullet points, using a numbered list is recommended.

Read Aloud

Due to the time and budget-sensitive nature of game development, it may be unlikely that your design documentation will ever be edited. In fact, I'm not sure I've ever been on a project where an editor copy edited an internal-facing design document. Your emails will certainly never be edited. As a result, it's important to get it as correct as possible on your own before handing it off to other developers or pressing the send button. One strategy I like is to reread what I've typed 'aloud.' I use quotes here because I personally don't actually read my writing out loud (although that's fine and probably a good idea), but I do slowly read it, a few times, without skimming, pronouncing all the words in my head and being careful to make sure it all makes sense and feels easy to read.

Avoid "Walls" of Text

I actively try to keep my paragraphs and blocks of text short. 'Walls' of text can intimidate a reader into the TLDR effect simply at a glance, before they even read the first word. To mitigate this, embrace new lines and white space. Use headings and bullet points as often as makes sense.

Consistent Style

Style is subjective, and more often times designers will find themselves responsible for defining the style of their documents. By style I mean things like: do you put a punctuation mark after a bullet point? Do you have one or two spaces after a period? Do you indent new paragraphs or simply use new lines? What font style and size do you use for headings? How about body text?

There often is no single correct answer. However, it's important to choose a solution and stick to it uniformly throughout your documentation.

The nature of game design leads to designers coining terms or naming official features and abilities. It's important to consistently address these things, and use the same style throughout all documentation. For example, consider *Magic: the Gathering*, where colors are part of the game design. If the designers capitalize the color words ("Black spells"), this should be done consistently throughout all the game's language. This goes for in-game text and promotional materials as well as documentation!

Avoid Technical Jargon and Extraneous Aesthetic Details

Game development is a creative endeavor, and the game industry is full of creative, talented people. When writing design documents, just describe how a feature should play out, gameplay-wise. Describe how things work from the player's perspective. The player has no clue how things are set up behind the scenes, and doesn't care. Avoid technical jargon. Don't tell programmers how to code the game, or artists how to draw it. They are professionals in their own field, and more than likely better at their craft than you. If an art or sound detail is important to your design or player usability (for example, you may want the explosive barrels to be red in order to read well to the player), then by all means include that detail in your documentation. Decide and agree on a general aesthetic vision and art style for the game. This should be documented with reference images and videos. However, leave any extraneous aesthetic details to the professionals hired to create them.

I recommend giving people a degree of creative freedom and working with your team to achieve some consensus on the art style and at least some game design elements. Especially if they are working on a volunteer basis. It helps people feel invested in your project, and the more excited they are about it and the more buy-in they have, the better the work usually turns out.

Avoid Marketing Language

It's very important to understand the difference between documenting a game and marketing a game. In documentation, you are speaking to other developers, helping them understand what

they must build. When marketing a game, you are speaking to a target audience of consumers, attempting to sell them your game and convincing them it's great.

Never use superfluous, subjective language such as "amazing levels" or "devious puzzles" in design documentation. You are not selling your game to developers, you are simply explaining it to them. If it sounds like it belongs on the back of the box or in a trailer, leave it out of your design document.

Images and Videos

Remember the old adage: a picture is worth a thousand words. If the goal is to use as few words as possible to describe a feature, then reference images and videos, diagrams, flow charts, storyboards, mock screenshots, UI wireframes, and other diagrams are highly useful.

You can search YouTube for gameplay videos with no commentary. If you have an extensive knowledge of existing games, this is where it pays off. Showing a programmer a similar feature in a gameplay video reference is incredibly effective.

Don't Write "You" When Referring to the Player or Player Character

When talking about games, it's common for people to refer to the player as "you." For example: "You have to face the Boo so it stops chasing you." That's fine for talking to your friends about a game, but when we're writing design documentation, it's best to be more technical, precise, and accurate:
"If the player character is facing a Boo, it freezes in place. If the player character is not facing the Boo, it chases the player."
You could also refer to the character by name, though this may be less accurate if players can choose from multiple avatars (e.g. Luigi): "If Mario faces a Boo…"

OLD DRAFT

Design Documents – Common Pitfalls

Students of game design sooner or later discover the notion of design documents: the blueprints or even ‘bible’ of a game describing exactly how it should function. Because this sort of thing

sounds so important to the act of game design, and the only real prerequisite is using a word processor, writing design documents seems to be a point of emphasis for many aspiring game designers.

I find this problematic for a few reasons. First, the written word, the exclusive domain of the design document, typically invites grand ideas without consideration of the cost of the features described in the document. Even when you try to limit your **scope**[1], without prior development experience or trying to prototype what you're describing, it's very likely to underestimate the cost of the features you're documenting.

Second, people can get hung up writing complex, detailed design documents. Trying to document every last element of a game idea before setting forth with development may seem like a noble pursuit, but there are some flaws with this idea. At some point, time is more effectively spent actually building the game rather than writing about it. What if the game idea you have spent so much time detailing has some fundamental flaw that can only be uncovered through playtesting? For this reason I recommend prototyping before getting too deep in documentation. Often times playing a game prototype uncovers the most fun additions, new directions, and best iterations you can make—as opposed to just closing your eyes and thinking about it. The earlier that design changes are made, the cheaper they are to implement. Late in a project's production, it is not uncommon for design changes to simply be off the table due to the expense and risk of implementing them. This means you must prototype and playtest your idea as early as possible to discover what you got wrong on paper, or simply overlooked.

As a result of the iteration process mentioned above, often times the details of a game design slowly shift (for the better) from their documented state. If you have chosen a "design document as bible" mantra, this means you must actively keep all of your documentation updated. In the throes of production, tweaks are made to games at a rapid pace. Keeping the documentation updated can become a consuming task in and of itself.

Additionally, the more attached you get to a document, the less open-minded you may become to changes. This becomes even more of a risk when only one designer writes the documentation before trying to convince a team to help them develop the game. Writing something down becomes a way to insulate and over-protect game ideas, but most of the time ideas should be malleable. Games require **iteration**[2] to become the best they can. If you are unwilling to make a change because it's not written that way in a document, your game will suffer and it's unlikely others will enjoy collaborating with you.

Finally, the longer and more intensely detailed your documentation becomes, the less likely anyone other than you is to read it. TLDR affects all professional communication, and that includes design documents. In the next section I'll offer some ideas for avoiding these pitfalls.

Design Document Strategies

It's important to realize that there's no single 'correct' way to document a game. In film, theatre, and TV, there are established standards for scripts and other planning documents, and deviating from them will quickly lead to your work being disregarded. But this isn't the case with game design documents, and there are no real industry standards.

I like to compare design documents to rough drafts in writing. Everyone has their own opinion about how to do rough drafts. Some people won't do them at all. Some insist on a very thorough rough draft, while others still find it most helpful to create an outline of what to write about before jumping in head first. I've worked with several different game studios and independent developers, and each has a different style and approach to their game design documentation. Through a bit of trial and error, you'll discover what works best for you.

While I won't recommend trying to fully detail every element of your game in a design document before you start building the game, documentation does play an important role in game development. Documentation helps developers stay on the same page, and provides important guidance for the implementation of features and the crystallization of your game's vision.

The important part is that those working on the game can find the relevant documentation easily, without scrolling endlessly through a 1000 page document. And once they do find the relevant section, they must understand it as closely to the designer's original intent as possible.

In the proceeding sections I will elaborate on some strategies for accomplishing these goals in your design documentation. And remember: if you're writing for an employer that expects documentation to fit a specific format, you'd better adhere to their standards!

What to Document with

If you're working on a large project, rather than creating a singular uber-document containing every detail of your game, I suggest creating a handful of shorter documents, adding new ones as necessary. Keeping a table of contents updated and managing a single huge document can soon spiral into a job in and of itself, which becomes quite a nuisance when you have other important tasks to finish on a deadline. Instead, creating a new document for each of your game's systems and sections is a sound approach. Similarly, creating a wiki with entries for each of these can also work well. The popular source control hosting website GitHub has built-in support for wiki documentation. But sometimes more powerful and complicated tools like wikis can add unnecessary complexity to reading and writing documentation, so don't be afraid to use a simple solution when it makes sense.

It's good practice to work from the cloud, using Google Documents or a similar service. This makes it easy for multiple collaborators to simultaneously edit the same document, and makes sharing with the rest of your team easy. This also makes it easy to work from multiple computers, in case you have a desktop and a laptop, etc. Further, it insures against any

computer mishaps that might befall you, whether it's a ruined hard drive or a power outage before you had a chance to save.

What to Document

First, it's a good idea to write a brief overview or summary. This is useful to initially acquaint the unfamiliar with the project, and give them some perspective of the big picture. Start with the basics:

- Title (or working title; don't get hung up on this!)
- What platforms are the game designed for? Mobile / touch screens? Gamepads?
- What's the genre of the game?
- What's the camera perspective?
- What / who is the player character?
- What's the object of the game?
- What are the controls?
- Example walkthrough of a gameplay section
    - Ever read a walkthrough guide for a game you're playing? They describe the setting of a section of the game, and list exactly what the player needs to do to progress. As a designer, writing something like this for a game that doesn't exist yet can help others understand the vision for the game.

Some of this info may make sense to write out in sentence format, others may work best as bullet points or headings.

Next, consider the main features of your game. How does player movement work? How does player health work? How does scoring work? How does the player progress through the game? What enemies or obstacles are there? What items or weapons are there? Are there any other special features in your game? Each of these make for good sections to add to your documentation.

Story != Game Idea

Many new designers tend to confuse story ideas for games for ideas for games themselves. Games can have stories, but stories and games are not the same thing. Interactivity is the defining characteristic of games, and your documentation should start by defining the interactive elements and making intended the player experience as clear as possible.

Later you can add sections of your design document that relate to the plot of the game, beginning with a high level outline.

Clarity: the Linchpin of Professional Communication

Design documents are technical writing and professional communication. As such, their most important quality is clarity. Your number one goal when writing in a design document should be to effectively, clearly communicate to your audience (that is, other developers responsible for implementing your design). An ideal design document communicates the designer's ideas so clearly that the party responsible for implementing it understands exactly what to do without ever speaking to the designer. This isn't always possible, or worth it, so sometimes it's best to just write down the basics with an understanding that you'll need to have a conversation with the other developers involved in implementing the feature before they start working on it. Consider a perfect instruction manual, either for a board game or some furniture you're trying to build. Ideally, it is short and sweet, and after reading it you understand exactly what to do. Here are some tips you can take to improve the clarity of your design documents.

TLDR

"To Long, Didn't Read" is a real thing. You should always keep it in the back of your mind when writing professional communication, whether sending quick work emails or authoring a design document. When I give my students reading assignments, I know that if it appears long, the odds of them reading it drop significantly.

A high word count is not only intimidating to readers, but it also dilutes your document, and makes it less likely that someone will easily notice something important. The odds of a detail getting overlooked go up with each new detail you add. For this reason, write as sparsely as possible and think carefully about what you choose to commit to documentation. Simplicity is tantamount to clarity. If you can rephrase or present something in a simpler way, do it. Remember the wisdom of Strunk and White: "Omit needless words," and "Avoid passive voice!"[3]

Short Sentences

In a similar spirit of avoiding TLDR, I also advise avoiding long, run-on sentences. The longer your sentences, the more confusing your writing becomes. And remember, our primary goals are simplicity and clarity! Consider the writing style of William Faulkner, notoriously difficult to read for its long, stream of consciousness-like, run-on sentences. This makes for great literature, but design documents should not aspire to be "artsy." They should be crystal clear, like the best instruction manual you've ever encountered. For these reasons, I recommend avoiding fancy punctuation and simply breaking compound sentences into separate, individual sentences. Another consideration: the longer a sentence, the more likely it is for something to go wrong. Each extra clause is a chance for a mistake or confusion. Simplicity is always best, and the fewer parts something has, the simpler it is.

Built-in Heading Styles

Use built-in heading styles as you write. This automates table of contents and outlining and helps ensure consistency in your formatting. Thinking about headings will help you break up your document into more digestible chunks.

Bullet Points

Along the same lines, sometimes complete sentences aren't even necessary for effective and concise communication. To an extent, I am a sucker for formality, but bullet points are perfectly fine for listing important information concisely. Just be wary of creating a list that's too long, or having a list go through a page break so that the bulleted items are no longer visible with their heading or introduction. If a list starts to grow out of control, perhaps there is a better way to catalog it, such as in a spreadsheet, or further subdivided into multiple sections.

Read Aloud

Due to the time and budget sensitive nature of game development, it may be unlikely that your design documentation will ever be edited. Your emails will certainly never be edited. As a result, it's important to get it as correct as possible on your own before handing it off to other developers or pressing the send button. One strategy I like is to reread what I've typed 'aloud.' I use quotes here because I personally don't actually read it out loud (although that's fine and probably useful), but I do slowly read it, a few times, without skimming, being careful to make sure it all makes sense and feels easy to read.

Avoid "Walls" of Text

I actively try to keep my paragraphs and blocks of text short. 'Walls' of text can intimidate a reader into the TLDR effect simply at a glance, before they even read the first word. To mitigate this, embrace new lines and white space. Use headings and bullet points as often as makes sense.

Consistent Style

Style is subjective, and more often times designers will find themselves responsible for defining the style of their documents. By style I mean things like: do you put a punctuation mark after a bullet point? Do you have one or two spaces after a period? Do you indent new paragraphs or simply use new lines? What font style and size do you use for headings? How about body text?

As I mentioned, these are subjective topics, and there often is no single 'correct' answer. However, it's important to choose a solution and stick to it uniformly throughout your documentation.

The nature of game design leads to designers coining terms or naming official features and abilities. It's important to consistently address these things, and use the same style throughout all documentation. For example, consider *Magic: the Gathering*, where colors are part of the game design. If the designers capitalize the color words ("Black spells"), this should be done

consistently throughout all the game's language. This goes for in-game text and promotional materials as well as documentation!

Don't Tell People How to Do Their Job

Game development is a creative endeavor, and the game industry is full of creative, talented people. When writing design documents, just describe how a feature should play out, gameplay-wise. Don't tell programmers how to program it, or artists how to draw it. They are professionals in their own field, and more than likely better at their craft than you. If it an art or sound detail is important to your design or player usability (for example, you may want the explosive barrels to be red in order to read well to the player), then by all means include that detail in your documentation. However leave any extraneous aesthetic details to the professionals hired to create them.

The exception to this is if you are personally paying people to make something for you. At that point, it's their job as contractors or employees to make exactly what you specify. But I still recommend giving people some creative freedom. It helps people feel invested in your project, and the more excited they are about it, the better it usually turns out.

Avoid Marketing Language

It's very important to understand the difference between documenting a game, and marketing a game. In documentation, you are speaking to other developers, helping them understand the technical details of what they must build. When marketing a game, you are speaking to a target audience of consumers, attempting to sell them your game.

Never use superfluous, subjective language such as "amazing levels" or "devious puzzles" in design documentation. You are not selling your game to developers, you are simply explaining it to them. If it sounds like it belongs on the back of the box or in a trailer, leave it out of your design document.

Images

Remember the old adage: a picture is worth a thousand words. If the goal is to use as few words as possible to describe a feature, pictures, flow charts, storyboards, mock screenshots, UI wireframes, and other diagrams are highly useful.

Don't Write "You" When You Mean "The Player"

When talking about games, it's common for people to refer to the player as "you." For example: "You have to face the Boo so it stops chasing you." That's fine for talking to your friends about a game, but when we're writing design documentation, it's best to be more technical, precise, and accurate:

"If the player character is facing a Boo, it freezes in place. If the player character is not facing the Boo, it chases the player."

You could also refer to the character by name, though this may be less accurate if players can choose from multiple avatars (e.g. Luigi): "If Mario faces a Boo…"

[1] **Scope** – the size and scale of your project. How long it will take to complete production. Scope is almost always underestimated, and many game projects suffer as a result, either being rushed out the door unfinished, or simply never seeing the light of day as developers realize they have bitten off more than they can chew.

[2] **Iteration –** the process of implementing an element of a game, playtesting it, and then adjusting it as necessary. Then of course the adjustment is tested, and further iterated on as necessary.

[3] Strunk and Write wrote *The Elements of Style*, a handbook for writing, popular in many US high schools. If you haven't read it, I highly recommend seeking it out, as the simple style rules will certainly improve your writing.