



Students get a Tuts+ subscription for just \$45! [Hurry limited offer.](#)

[Dismiss](#)

All Topics

Find tutorials, courses



Free Tutorials Courses

eBooks

Pricing Sign In

Free Account

Safari Power Saver
Click to Start Flash Plug-in



Advertisement



Game Development

Categories

Learning Guides

PROJECT MANAGEMENT

How (and Why) to Write a Great Game Design Document

by [Alex Sayenko](#) 3 days ago 2 Comments

302 106 17

Every indie developer or team has asked themselves how best to manage the development process. Is it obligatory to use detailed documentation, such as the legendary game design document (GDD)? What are the most common mistakes, and how can they be avoided?

For those who have searched for answers to these questions, I want to share our team's experience of creating our GDD.

Why You Need a GDD

Good Documentation is a Must

Many game designers abstain from cultivating design documents and organizing ideas. I can't imagine how these developers get through those times that they are hampered by malfunctioning code, placeholder art, and clashing mechanics, all while trying to remember why it was that they first subjected yourself to making a game in the first place.

Having a well thought out game design document can act as your crutch in these times. It will let you see the awesome ideas and concepts that kept you up all night long when you first started on your game, and, perhaps more importantly, which ones need to be cut from the game to make your life easier, or reworked to make your efforts worthwhile.

A GDD Aids Game Design and Development

A game design document acts as a nexus and hub to connect and list every aspect of a game. It consists of written descriptions, images, graphs, charts and lists of information pertinent to each segment of development, and is often organized by what features will be in the game, and clearly lays out how they will all fit together.

Creating a GDD will aid your team's designer in understanding what the essence of the game is and the planned scope of its overarching world and gameplay. Plus, having all the game elements in one well-organized document will help the designer easily convey their vision to the rest of the team, while also helping to pinpoint weaknesses or missing components that the game may require.

The GDD should serve as your master checklist. It will be the document you toss up in the air in celebration upon completing all its sections and finishing your game.

A GDD Helps in Other Areas, Too

Since a GDD is full of descriptions, it is an ideal resource for all PR and marketing fronts, with concepts that convey the game's aesthetic and appeal already written up and ready to copy and paste.

Prospective employees' skills can be quickly assessed as qualified or not by looking at their credentials alongside their positions' corresponding sections in the document.

If fundraising is in the cards for your team, it will be crucial to have a well put together GDD for investors to look over and determine the risks of your development, as well as your ability to deliver on your promises.

A GDD Keeps Everyone on the Same Page

Creating and adhering to a game design document is like planting a seed and watching it grow into a tree over the course of development. You have your initial preparation, cultivation, and ultimately the gruesome and backbreaking toil of the harvest.

A common mistake is not handing all your team members the proper [gardening tools](#) to make your game a reality. The GDD will help make sure everyone is working together, so that you don't find your programmer and artist cutting off the branch they are standing on.

Common Mistakes

Trying to Describe Everything at Once

It is not necessary to list every feature and mechanic in detail in the first draft. This is impossible when working on a complex game with a small team. Outlining the major nodes of gameplay and core elements will help give you perspective on what needs to get done, but you should expect to fill them in one by one, over time.

Not Setting Deadlines

Setting game goals with deadlines may seem off-putting to some developers. Deadlines are a part of the boring 9-to-5 lifestyle, so many people have a natural aversion to them.

However, setting deadlines is how you ensure that your game will get made and not sit forever unfinished in a folder on your desktop. These types of goals are milestones on the road to progress, and passing them one by one is a clear indication that you are doing something right. Each is a cause for celebration.

Deadlines are a basic component of scheduling that helps you monitor the performance of your team and yourself. It aids in decision making that is rooted in reality, and eventually builds up a momentum and ethic that is healthy for the team as a whole and the individuals within it.

Assuming Everyone Knows What to Add

There is room in the GDD for basic descriptions that cover gameplay, storylines, and major coding tasks. As development progresses, you add more details to these sections. While creating and testing the game, you should add or update specific technical details every time a feature is implemented or changed. This way, you'll never have a build with elements that you can't trace back to the GDD, creating missing links of ideas, code or art. It can also be helpful to add information about the difficulty of implementing certain tasks, and whether they have been fully integrated into the game or may require further revision.

Throughout this process, your initial concepts should be trickling down into ever more detailed descriptions of each facet your features contain. This helps make concrete milestones that are easy to navigate and backtrack to see where they originated and where they need to go.

As the GDD continues to grow and become more finalized it is still important to keep it up to date. This eliminates situations where team members do something without being able to justify why they spent time doing them, which is crucial come crunch time.

Printing the GDD

Personally, I have an unexplained primal fear of drowning in a heap of printed documentation. This would become a real nightmare if I had to keep all old versions of our GDD for every team member.

Why should we torture ourselves like this in the age of digital technology? There are plenty of free online services like [Google Docs](#) or [Trello](#) that let you save all changes and see all your team's comments in real time.

How to Write an Effective GDD

Write It in Stages

When starting the GDD, it is normal to get wrapped up in concepts. Backgrounds, introductions, and key descriptions help to flesh out the game and give it shape. As you begin to test and implement features, these concepts should become more refined, specified and detailed. Maintaining proper organization will become more and more critical as your GDD gains weight and density.

Start in the concept phase, where you brainstorm your ideas and get them all down on paper. This should be exciting! It will also serve as a roadmap so you don't lose track of your goals and vision along the way. When the appeal of certain game elements lose their lustre or lead you into a ditch, it may be time to rework your initial concept to make sure you reach a satisfying finish line.

Towards the middle of development, once you have a gung-ho team on board, discussions and game builds will help sculpt and organize the document into an easy to use and hefty guide for everyone. There's still room for experimenting with new concepts and ideas at this point, but they should be kept in check with some of your initial documentation.

The home stretch of development is where your game design document will save you hours of frustration and heartache. As you close in on release your GDD should slowly start turning into a stone tablet, with features and mechanics set in more permanent game builds all held together by art that was surely crafted over multiple iterations to match the document's specifications. The document should help keep all the team's wheels on the ground, with a good line of sight on realistic expectations on delivering the game.

It is not necessary to have an absolutely complete GDD before starting development. But the GDD *must* be complete for the next 10 days or two weeks beyond your team's current work—and the relevant parts of the document must be as detailed as possible.

Allow Changes During Your Game's Development

Parts of the GDD will have to be changed and modified throughout the entire development process, sometimes even in the final days before release. It can start to resemble a disaster zone if the content is not trimmed properly. If you are afraid of deleting text that is outdated, cut and paste it into an addendum or separate document. This will leave the main body of the GDD relevant to the current state of development, without all the distractions of previous iterations.

Don't ever stop team members submitting new ideas. Idea creation is one of the most rewarding parts of development, and should be encouraged at all times. Your team members should head into development knowing that many of these concepts will be cut and never make it into the game, but this shouldn't stop them from dreaming! No one knows what ideas will produce the best results at first, so generating new and innovative ideas should be a staple of your discussions and celebrated accordingly.

Put Just One Person in Control of It

Supervision of the GDD must be performed only by one team member. They will identify the key ideas that must be focused on, and cut the less important ideas.

Encouraging active feedback is important, then, because other team members do not have the chance to add their ideas to the document directly.

Most development issues are comprised of a hard outer shell of miscommunication and a soft interior of not knowing how to compensate and correct them. These barriers can be eliminated with vigilant maintenance of the GDD and clear, concise documentation, and this can best be achieved if one person takes on that responsibility.

Focus on Readability

Be consistent with font styles and using uniform headers and indentations, punctuation, and formatting. Creating a legend or key to explain what specific colored highlights mean can go a long way toward reducing confusion and cutting down on the time it takes to convey the stages of different features' implementation.

Keep the Language Clear

The simpler and clearer you keep the language in the GDD, the better your team will understand it.

It's important to keep the writing clear and concise, and your team should actively give you feedback about the presentation and clarity of the GDD. A back-and-forth dynamic will result in a more cohesive development experience, with overarching benefits including a defined art style, fewer communication errors, and less stressful documentation and office work.

But most importantly, the GDD should be a reflection of your team culture, created in whatever format you find works best and is most appealing to you and those you work with.

Use Visual Aids

No one should ever be able to say that they haven't understood something, or done something correctly, due to a lack of reference material in the GDD.

Visual materials and references play a key role in the process of conveying ideas. Some difficult concepts can be explained in less time with visual aids like graphs and concept art. This will help ensure every team member understands the information that is conveyed to them, which in return will help them to complete development tasks faster.

Put Some Passion Into It

You should not restrict yourself to dry text. (If you do, you'll be waiting a long time for everyone to get engaged and understand the main ideas!) Try to describe the players' emotions and the experiences that the game might cultivate.

Keeping a GDD may sound technical, but you shouldn't be afraid to tear your heart out and throw it at the document. Let your emotion and passion bleed into it. Imagine how you want to make the player feel, and write those aspirations down alongside the descriptions of your features. This helps to cultivate a collective consciousness in your team about what your game is trying to convey to the player—and, let's face it, feelings should have enthusiasm behind them if you want them to be understood by anyone else.

Use It to Keep People on Track

Set the priorities of tasks and features, document their deadlines, and control their execution. You can't develop absolutely all the ideas which your team and your mind will propose, so (after you've cut some of them), you need to set their priorities and at least approximate a schedule for their implementation.

A well-groomed GDD makes for an excellent, prioritized list of tasks that need to be completed by your team. Not all the features in a GDD will make it into the final game. With this in mind, you must decide which features take precedence over others, and you should schedule them for implementation and testing before those others.

Think hard about what is critical for your game, and what is possible given your team's skill level, and use that information to guide your production.

A solid GDD can also help ease new team members into the project, and get them just as excited about it as you are.

Since a fully outlined GDD may result in what seems like an excessively daunting game to make, it is good to remember that more than one person will be fleshing out its specifics. Assigning your team members tasks in the GDD will help it to become more robust while keeping everyone on the same page. Anyone can jump in to the document and see what has been completed, what tasks lay before them and the rest of the team, and why they are working on their current task.

Continue Having Design Discussions

Writing something in a shared GDD shouldn't minimize or eliminate discussion with the team, it should serve to increase team discussion and improve your communication dynamics. It's important that everyone clearly understands how you (and the other team members) imagine each feature of the game.

Cutting ideas can be difficult and unnerving, but it is a process innate to creating games. Making sure that open, free discussion is a part of development will help ease the inherent tensions here, without dissuading members from being creative.

Play the Game In Your Mind

I found many good ideas virtually playing the game in my mind, both before and during the game's creation. Of course, this doesn't give any guarantee that those ideas will take roots in the game during development and testing but, especially in the early stages, it's a good method of brainstorming.

Set Realistic Goals

While it is good to foster an air of excitement within a team, it is equally important to keep your game goals embedded in reality. Mechanics and complex enemy and level behaviors always look great on paper, but reality has a way of corroding the grandeur of certain game elements, and this should be expected.

Consequences of updates and changes are almost impossible to foresee ahead of time in some situations, so just remember it is your job to try and limit the amount of remodelling that needs to be done when changes arise. If you make it common practice to play through new ideas in your mind before putting them down on paper, you stand a greater chance of keeping development goals rooted in realistic expectations.

Make Use of Free Online Tools

Our team is multinational. We live around the world, in different time-zones, and this makes it impossible to use print versions of documents for everyone, and difficult to have real-time conversations. Using tools such as [Skype](#) (for conversations), [Google Drive](#) (for sharing files), [Google Docs](#) (for collaborating on documents, and sharing the GDD), and [FlockDraw](#) (for digital drawings) can really help with explanations and

discussions.

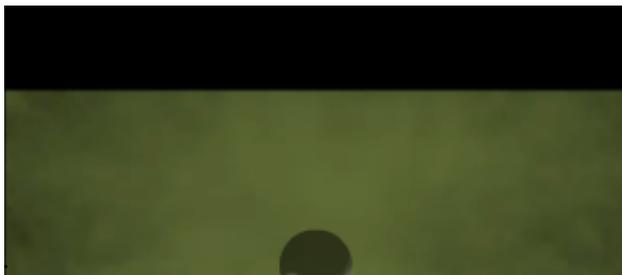
Conclusion

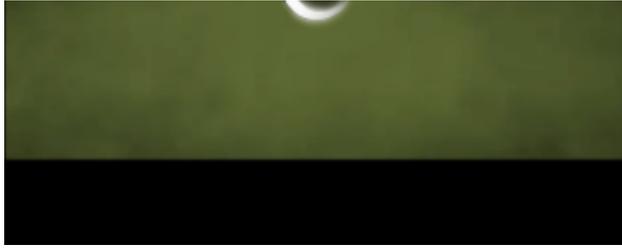
If you find yourself on the fence about whether maintaining a GDD is necessary for your game's production, you should take a good, hard look at how you envision development. There are almost certainly times where real life and full time jobs will get in the way of game making, or your implementation of features and mechanics simply does not work out.

On the stormy seas of game development, a healthy GDD can serve as a sturdy and solid vessel, and even a lifeboat at times. It is a detailed journal of your struggles and triumphs, a collection of thoughts and ideas for you to fall back on during hard times. You might find that improving the quality of the GDD trickles down into the rest of development as well, raising the bar across the board for your team. It should serve as a sturdy hub to facilitate team discussion and generate new, even greater ideas. At the same time, it can keep these concepts in realistic check.

The benefits of this type of efficiency may seem small when looked at individually, but over the long course of development, it builds up a wonderful kind of momentum. Ultimately, this type of document should propel, compel, and inspire you and your team to finish what you started. It should show you that your game has a plan that can be realized.

And after you have finished your game, your GDD will stand as a testament to all of your hard work and efforts, the behind-the-scenes of an elaborate experience to be enjoyed by all.





Advertisement

Recommended References and Tools

- [Effectively Organize Your Game's Development With a Game Design Document](#) by Gamux
- [Developing Your Game Concept By Making A Design Document](#) by Mare Kuntz aka sunandshadow
- [The Anatomy of a Design Document, Part 1: Documentation Guidelines for the Game Concept and Proposal](#) by Tim Ryan
- [GAME DESIGN 101: "I have a Great Idea for a video game... how do I sell it and get rich and famous?"](#) by Tom Sloper
- [Creating A Great Design Document](#) by Tzvi Freeman
- [Documentation for Indie Studios: Why do you need it?](#) by Ivan Spasov
- [FlockDraw](#): Draw in real time with others
- [Enloop](#): Free online business plans
- [Trello](#): Simple project management
- [GameDev Resources spreadsheet](#) compiled by David Arcila

References

- Preview image: [Documents](#) by James Cook
- Preview image: [Skate Park](#) by Juan Pablo Bravo





Advertisement

Categories:

- Project Management
- Planning
- Documentation
- GDD
- Teamwork



Translations Available:

Tuts+ tutorials are translated by our community members. If you'd like to translate this post into another language, [let us know!](#)

About Alex Sayenko



My name is Alex.



Advertisement

Related Tutorials



Making AI Fun: When Good Enough is Good Enough

Game Development



Bringing Your Game to Life in 10 Simple Steps

Game Development



To the Point: Interview With Rod Hunt

Design & Illustration

Jobs



Digital Designer UX/UI
at La Moulade in London, UK



Full Stack Developer
at La Moulade in London, UK

Envato Market Item

Original Hover Effects

- 1 html structure
- 3 layout
- 27 Hover Effects

Icons: Desktop, Tablet, Smartphone

Website Name: Tintung

2 Comments Tuts+ Hub

1 Login ▾

Recommend 1 Share

Sort by Best ▾



Join the discussion...



Alex Sayenko · a day ago

Thanks for such kind words. Russian version of the article is in the progress right now. So, it will be soon.

^ | ▾ · Reply · Share ›



DmitryGh · a day ago

I think is nice & useful! :) Have you this article in another language, in russian, for example?

^ | ▾ · Reply · Share ›

Subscribe

Add Disqus to your site

Privacy



Advertisement

Teaching skills to millions worldwide.

19,506 Tutorials

520 Video Courses

Follow Us



Email Newsletters

Get Tuts+ updates, news, surveys & offers.

Email Address

Subscribe

Privacy Policy

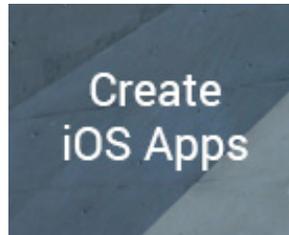
Help and Support

- FAQ
- Terms of Use
- Contact Support
- About Tuts+
- Advertise
- Teach at Tuts+



Custom digital services like logo design, WordPress installation, video production and more.

[Check out Envato Studio](#)



Browse 500+ unique template files for iOS apps. Create games and utility apps with pre-built app code.

[Browse iOS Apps on CodeCanyon](#)

