7 Writing a Game Design Document

> If you don't know where you are going. How can you expect to get there?
>
> *Basil S. Walsh*

One of the most discussed things in the world of Game Development is the so-called "GDD" or "Game Design Document". Some say it's a thing of the past, others swear by it, others are not really swayed by its existence.

Being an important piece of any software development process, in this book we will talk about the GDD in a more flexible way.

7.1 What is a Game Design Document

The Game Design Document is a Body Of Knowledge that contains everything that is your game, and it can take many forms, such as:

- A formal design document;
- A Wiki$_g$;
- A Kanboard $_g$.

The most important thing about the GDD is that it contains all the details about your game in a centralized and possibly easy-to-access place.

It is not a technical document, but mostly a design document, technical matters should be moved to a dedicated "Technical Design Document".

7.2 Possible sections of a Game Design Document

Each game can have its own attributes, so each Game Design Document can be different, here we will present some of the most common sections you can include in your own Game Design Document.

7.2.1 Project Description

This section is used to give the reader a quick description of the game, its genre (RPG, FPS, Puzzle,...), the type of demographic it covers (casual, hardcore, ...). Additional information that is believed to be important to have a basic understanding of the game can be put here.

This section should not be longer than a couple paragraphs.

A possible excerpt of a description could be the following:

> This game design document describes the details for a 2D side scrolling platformer game where the player makes use of mechanics based on using arrows as platforms to get to the end of the level.
>
> The game will feature a story based on the central America ancient culture (Mayan, Aztec, ...).
>
> The name is not defined yet but the candidate names are:

7.2.2 Characters

If your game involves a story, you need to introduce your characters first, so that everything that follows will be clear.

A possible excerpt of a characters list can be the following:

> **Ohm** is the main character, part of the group called "The Resistance" and fights for restoring the electrical order in the circuit world.
>
> **Fad** is the main side character, last survivor and heir of the whole knowledge of "The Capacitance" group. Its main job is giving technical assistance to Ohm.

> **Gen. E. Rator** is the main antagonist, general of "The Reactance" movement, which wants to conquer the circuit world.

This can be a nice place where to put some character artwork.

If your game does not include a story, you can just avoid inserting this section altogether.

7.2.3 Storyline

After introducing the characters, it's time to talk about the events that will happen in the game.

An example of story excerpt can be the one below:

> It has been 500 mega-ticks that the evil **Rator** and the reactance has come to power, bringing a new era of darkness into the circuit world.
>
> After countless antics by the evil reactance members, part of the circuit world's population united into what is called "The Resistance".
>
> Strong of thousands of members and the collaboration of *the Capacitance*, the resistance launched an attack against the evil reactance empire, but the empire stroke back with a carpet surcharge attack, decimating the resistance and leaving only few survivors that will be tasked to rebuild the resistance and free the world from the reactance's evil influence.
>
> This is when a small child, and their parents were found. The child's name, **Ohm**, sounded prophetic of a better future of the resistance.
>
> And this is where our story begins.

As with the Characters section, if your game does not include a story, you can just skip this section.

7.2.3.1 The theme

When people read the design document, it is fundamental that the game's theme is quickly understood: it can be a comedy-based story, or a game about hardships and fighting for a better future, or maybe it is a purely fantastic game based on ancient history…

Here is a quick example:

> This is a game about fighting for a better future, dealing with hardships and the deep sadness you face when you are living in a world on the brink of ruin.
>
> This game should still underline the happiness of small victories, and give a sense of "coziness" in such small things, even though the world can feel cold.

If you feel that this section is not relevant for your game, you can skip it.

7.2.3.2 Progression

After defining the story, you should take care of describing how the story progresses as the player furthers their experience in a high-level fashion.

An example:

> The game starts with an intro where the ruined city is shown to the player and the protagonist receives their magic staff that will accompany them through the game.
>
> The first levels are a basic tutorial on movement, where the shaman teaches the player the basic movement patterns as well as the first mechanic: *staff boosting*. Combat mechanics are taught as well.
>
> After the tutorial has been completed, the player advances to the first real game area: **The stone jungle**.
>
> …

7.2.4 Levels and Environments

In this section we will define how levels are constructed and what mechanics the will entail, in detail.

We can see a possible example here:

> The First Level (Tutorial) is based in a medieval-like (but adapted to the center-America theme) training camp, outside, where the player needs to learn jumping, movement and fight straw puppets. At the end of the basic fighting and movement training, the player is introduced to *staff boosting* which is used to first jump to a ledge that is too high for a normal jump, and then the mechanic is used to boost towards an area too far forward to reach without boosting.
>
> ...

Some level artwork can be included in this section, to further define how the levels will look and feel.

7.2.5 Gameplay

This section will be used to describe your gameplay. This section can become really long, but do not fear, as you can split it in meaningful sections to help with organization and searching.

7.2.5.1 Goals

> Why is the player playing your game?

This question should be answered in this section. Here you insert the goals of your game, both long and short term.

An example could be the following:

> Long Term Goal: Stop the great circuit world war
>
> Optional Long Term Goal: Restore the circuit world to its former glory.
>
> Short Term Goals:
>
> - Find the key to the exit
> - Neutralize Enemies
> - Get to the next level

7.2.5.2 Game Mechanics

In this section, you describe the core game mechanics that characterize the game, extensively. There are countless resource on how to describe game mechanics, but we'll try to add an example here below.

> The game will play in the style of the well-known match-3 games. Each match of 3 items will add some points to the score, and new items will "fall" from a randomly chosen direction every time.
>
> Every time an "L" or a "T" match is performed, a special item of a random color will be generated, when a match including this item is made, all the items in the same row and column will be deleted and bonuses will be awarded.
>
> Every time a match with 4 items in a row is performed, a special item of a random color will be generated, when a match including such item is made, all items in a 3x3 grid centered on the item will be deleted and bonuses will be awarded.
>
> Every time a match with 5 items in a row is performed, a special uncolored item will be generated, this can be used as a "wildcard" for any kind of match.
>
> In case the 5-item special is matched with any other special item, the whole game board will be wiped and a bonus will be awarded.
>
> ...

7.2.5.3 Skills

Here you will describe the skills that are needed by the users in order to be able to play (and master) your game.

This will be useful to assess your game design and eventually find if there are some requirements that are too high for your target audience; for instance asking a small child to do advanced resource management could be a problem.

This will also help deciding what the best hardware to use your game on could be, for instance if your game requires precise inputs for platforming then touch screens may not be the best option.

Here's an example of such section:

> The user will need the following skills to be able to play the game effectively:
>
> - Pressing Keyboard Buttons or Joypad Buttons
> - Puzzle Solving (for the "good ending" overarching puzzle)
> - Timing inputs well (for the sections with many obstacles)
>
> ...

7.2.5.4 Items/Powerups

After describing the basic game mechanics and the skills the user needs to master to be able to play the game effectively, you can use this section to describe the items and powerups that can be used to alter the core gameplay.

For example:

> The player can touch a globular light powerup to gain invincibility, every enemy that will touch the player will get automatically killed. The powerup duration is 15 seconds.
>
> Red (incendiary) arrows can be collected through the levels, they can get shot and as soon as they touch the ground or an enemy, the burst into flames, similarly to a match.
>
> ...

In this section you describe all items that can be either found or bought from an in-game store or also items derived from micro-transactions. In-game currency acquisition should be mentioned here too, but further detailed in the monetization section.

7.2.5.5 Difficulty Management and Progression

This section can be used to manage how the game gets harder and how the player can react to it. This will expand on game mechanics like leveling and gear.

This section is by its own nature quite subjective, but describing how the game progresses helps a lot during the tighter parts of development.

Below a possible example of this section:

> The game will become harder by presenting tougher enemies, with more armor, Health Points and attack. To overcome this difficulty shift, the player will have to create defense strategy and improve their dodging, as well as leveling up their statistics and buy better gear from the towns' shops.
>
> In the later levels, enemies will start dodging too, and will also be faster. The player will need to improve their own speed statistic to avoid being left behind or "kited" by fast enemies.
>
> As the game progresses, the player will need to acquire heavy weapons to deal with bigger bosses, as well as some more efficient ranged weapons to counteract ranged enemies.
>
> ...

This section is good if you want to talk about unlocking new missions/maps/levels too.

7.2.5.6 Losing Conditions

Many times we focus so much on how the player will get to the end of the game that we absolutely forget how the player can *not* get to the end of the game.

Losing conditions must be listed and have the same importance of the winning conditions, since they add to the challenge of the game itself.

A possible example of how a "losing conditions" section could be written is the following:

> The game can be lost in the following ways:
>
> - Losing all the lives and not "continuing" (Game Over)
> - Not finding all the Crystal Oscillators (Bad Ending)

A possible variation on the theme could be having an "endings" section, where all (both good, bad and neutral) endings are listed.

7.2.6 Graphic Style and Art

Here we describe the ideas on how the game will look like. Describing the graphic style and medium.

Here is a possible example of the game:

> This is a 2D side scroller with a dark theme, the graphics should look gloomy and very reminiscing of a circuit board.
>
> The graphical medium should be medium-resolution pixel art, allowing the player's imagination to "fill in" the graphics and allowing to maintain a "classic" and "arcade" feeling.
>
> ...

7.2.7 Sound and Music

Sadly, in way too many games, music and sound is an afterthought. A good soundtrack and sound effect can really improve the immersion, even in the simplest of games.

In this section we can describe in detail everything about Music and Sound Effects, and if the section becomes hard to manage, splitting it in different sub-sections could help organization.

> Music should be based on the glitch-hop style, to complement the electronic theme. 8 or 16-bit style sounds inside the score are preferable to modern high-quality samples.
>
> Sound effects should appeal to the 8 or 16-bit era.
>
> Lots of sound effects should be used to give the user positive feedback when using a lever to open a new part of the level, and Extra Lives/1UP should have a jingle that overrides the main music.

7.2.8 User Interface

In this section we will describe everything that concerns the User Interface: menus, HUD, inventories and everything that will contribute to build the user experience that is not strictly tied to the gameplay.

This is especially important in games that make heavy use of menus, like turn-based strategy games or survival games where inventory management can be fundamental.

Let's see an example of how this section can be written:

> The game will feature a cyberpunk-style main menu, looking a lot like an old green-phosphor terminal but with a touch of futurism involved. The game logo should be visible on the left side, after a careful conversion into pixel-art. On the right, we see a list of buttons that remind old terminal-based GUIs. On the bottom of the screen, there should be an animated terminal input, for added effect.

> Every time a menu item is highlighted or hovered by the mouse, the terminal input will animate and write a command that will tie to the selected menu voice, such as:
>
> - Continue Game: `./initiate_mission.bin -r`
> - Start Game: `./initiate_mission.bin --new`
> - Options: `rlkernel_comm.bin --show_settings`
> - Exit: `systemcontrol.bin --shutdown`
>
> The HUD display should remind a terminal, but in a more portable fashion, to better go with the "portability" of a wrist-based device.

It's a good idea to add some mock designs of the menu in this section too.

7.2.9 Game Controls

In this section you insert everything that concerns the way the game controls, eventually including special peripherals that may be used.

This will help you focusing on better implementing the input system and limit your choices to what is feasible and useful for your project, instead of just going by instinct.

Below, a possible way to write such section

> The game will control mainly via mouse and keyboard, using the mouse to aim the weapon and shoot and keyboard for moving the character.
>
> Alternatively, it's possible to connect a twin-stick gamepad, where the right stick moves the weapon crosshair, while the left stick is used to move the character, one of the back triggers of the gamepad can be configured to shoot.
>
> If the gamepad is used, there will be a form of aim assistance can be enabled to make the game more accessible to gamepad users.

7.2.10 Accessibility Options

Here you can add all the options that are used to allow more people to access your game, in more ways than you think.

Below, we can see an example of many accessibility options in a possible game.

> The game will include a "colorblind mode", allowing the colors to be colorblind-friendly: such mode will include 3 options: Deuteranopia, Tritanopia and Monochromacy.
>
> Additionally, the game will include an option to disable flashing lights, making the game a bit more friendly for people with photosensitivity.
>
> The game will support "aim assistance", making the crosshair snap onto the enemy found within a certain distance from the crosshair.
>
> In order to assist people who have issues with the tough platforming and reaction times involved, we will include the possibility to play the game at 75%, 50% and 25% speed.

7.2.11 Tools

This section is very useful for team coordination, as having the same toolkit prevents most of the "works for me" situations, where the game works well for a tester/developer while it either crashes or doesn't work correctly for others.

This section is very useful in case we want to include new people in our team and quickly integrate them into the project.

In this section we should describe our toolkit, possibly with version numbers included (which help reducing incompatibilities), as well as libraries and frameworks. The section should follow the trace below:

> The tools and frameworks used to develop the game are the following:
>
> **Pixel Art Drawing:** Aseprite 1.2.13
>
> **IDE:** Eclipse 2019-09
>
> **Music Composition:** Linux Multimedia Studio (LMMS) 1.2.1
>
> **Map and level design:** Tiled 1.3.1
>
> **Framework:** SFML 2.5.1
>
> **Version Control:** Git 2.24.0 and GitLab

7.2.12 Marketing

This section allows you to decide how to market the game and have a better long-term plan on how to market your game to your players.

Carefully selecting and writing down your target platforms and audience allows you to avoid going off topic when it comes to your game.

7.2.12.1 Target Audience

Knowing who is your target audience helps you better suit the game towards the audience that you are actually targeting.

Here is an example of this section:

> The target audience is the following:
>
> Age: 15 years and older
>
> Gender: Everyone
>
> Target players: Hardcore 2D platformer fans

7.2.12.2 Available Platforms

Here you describe the launch platforms, as well as the platforms that will come into the picture after the game launched. This will help long term organization.

Here is an example of how this section could look:

> Initially the game will be released on the following platforms:
>
> - PC
> - Playstation 4
>
> After launch, we will work on the following ports:
>
> - Nintendo Switch
> - XBox 360
>
> After working on all the ports, we may consider porting the game to mobile platforms like:
>
> - Android 9.0 +
> - iOS 11.0 +
>
> ...

7.2.12.3 Monetization

In this optional section you can define your plans for the ways you will approach releasing the game as well as additional monetization strategies for your game.

For example:

> The game will not feature in-game purchases.
>
> Monetization efforts will be focused on selling the game itself at a full “indie price” and further monetization will take place via substantial Downloadable Content Expansions (DLC)
>
> The eventual mobile versions will be given away for free, with advertisements integrated between levels. It is possible for the user to buy a low-price paid version to avoid seeing the advertisements.

7.2.12.4 Internationalization and Localization

Internationalization and Localization are a matter that can make or break your game, when it comes to marketing your game in foreign countries.

Due to political and cultural reasons, for instance you shouldn’t use flags to identify languages. People from territories inside a certain country may not be well accepting of seeing their language represented by the flag of their political adversaries.

Another example could be the following: if your main character is represented by a cup of coffee, your game could be banned somewhere as a “drug advertisement”.

This brings home the difference between “Internationalization” and “Localization”:

Internationalization Making something accessible across different countries without major changes to its content
Localization Making something accessible across different countries, considering the target country’s culture.

We can see a possible example of this section below:

> The game will initially be distributed in the following languages:
>
> - English
> - Italian
>
> After the first release, there will be an update to include:
>
> - Spanish
> - German
> - French

7.2.13 Other/Random Ideas

This is another optional section where you can use as a “idea bin”, where you can put everything that you’re not sure will ever make its way in the game. This will help keeping your ideas on paper, so you won’t ever forget them.

We can see a small example here:

> Some random ideas:
>
> - User-made levels
> - Achievements
> - Multiplayer Cooperative Mode
> - Multiplayer Competitive Mode