Serious Game Design Document Outline

A game design document is the blueprint from which a game is to be built. As such, every single detail necessary to build the game should be addressed. The larger the team and the longer the design and development cycle, the more critical is the need. For your purpose, the intent is to capture as much as possible of your design. I want you to think big…bigger than what you are able to develop. I also want you to be clear about what the software delivers and what the design entails. Use this document to define the ultimate game, but be clear about what you have delivered.

This document also includes elements that are normally considered part of a software design document but are included to give a more complete picture of the game.

You do not need to include all topics and you do not need to use the numbered outline structure for your document., but as a team you should review all items to be sure that you have considered all of these elements and why they are or are not relevant to your game. It should also be easy for me to identify the topic that you are covering.

1. Title Page (page and all elements are required)
   - Game Name – This is more important than you think even in a non-commercial environment. It sets the tone for the game's use and gives it an identity.
   - Tag line – a tag line is a statement or phrase that brands your game. Famous tag lines are "Just do it! "Got milk?" "The King of Beers" Or from the video game world: Dark Souls: "Prepare to Die", Dead Space: "Only the Dead Survive", Deus Ex "Trust no one. Question Everything", Pokémon: "Gotta catch em all!"
   - Team – names and roles. Remember that your client is part of the team
   - Date of last update
2. Revision History – only needed if you are working on multiple versions of the document or making major changes to the game. Probably not needed for most games in this class. However if I review a version that then significantly changes, include the history.
3. Game Overview (required of all games) – this is intended as a cohesive paragraph or two that lets someone understand what you are working on. It must include
   - Purpose of the Game – pedagogical objective(s) for educational games, training goal, or intended social impact. If it is addressing a problem, explain what that problem is (for example, students have a hard time …)
   - Intended Use – How the game is to be inserted into a pedagogical scenario or how will it be used in training, therapy, rehabilitation, etc. You should distinguish between a game that is stand alone, a supplement to a class, or working with an instructor
   - Justification for the use – Is there research that shows that this is apt to be successful? Are there are other games or applications that take a similar approach? If this is an experiment to see if it will work, why does the researcher believe that it might?
   - Target audience. There's a significant difference in a game teaching biology to a kindergartener or to a college student
   - Genre(s)

4. Gameplay – this section would tell a player what to expect. Not a lot of detail (that comes later).
   - Objectives – What are the objectives of the game? Is it to reach a destination, retrieve the chalice, accrue the most money, or solve a puzzle in the fastest amount of time?
   - Game Progression and Play Flow – How does the game flow for the game player? Is there an overall scoring and objective for the game or is it a progression of levels?
   - Mission, challenge or puzzle structure
5. Mechanics (Key Section) -- These are topics that should be covered if relevant. All games are different and not everything applies. Consider this a checklist to help you capture all that is important. It should be complete enough that someone could build a clone of your game based on this detail.
   - Rules – include both implicit and explicit rules.
   - Model of the game universe. Think of it as a simulation of a world, how do all the pieces interact? If it is a puzzle game, there may be no universe to discuss, but there is the layout and the structure of the puzzle. Included in the universe are

     Physics – How does the physical universe work? If there are no physics, do not try to create them.

     Economy – What is the economy of the game? How does it work? The economy incudes the items that the player is trying to collect as well as a world economy. How does a player earn or lose these items?
   - Character actions. These include

     Character movement --How does the character move?

     Objects – how to pick them up and move them. Are there key objects of importance that should be identified here? Is there an inventory that the player maintains or are all interactions transitory?

     Actions – What else can the player (or character) do? Include any in-game switches and buttons that are used, interacting with objects, and what means of communication are used

     Combat – If there is combat or even conflict, how is this specifically modeled? What are the differences in weapons?
   - Screen Flow -- How each screen is related to every other and a description of the purpose of each screen
   - Game Options - What are the options and how do they affect game play?
   - Replaying and saving
   - Cheats and Easter Eggs if there are any
6. Story and Narrative -- If you have a puzzle game, there is probably no backstory or story. As soon as you add an avatar of any sort, you have a story. It may be as simple as who the character is.
   - Backstory. If there is no plot or narrative in the story but you need to understand the background, that is a backstory. The backstory can also be setting the stage for the story that is going to unfold.
   - Plot elements. There are multiple patterns of a plot: the 3-act play, the hero's journey, episodic, … Use one of the models to describe the story of the game.
   - Game story progression. Explain how the progression of the plot is exposed to the player. Remember that there are different approaches to the narrative: how it is exposed, who is

telling it and the sequence that things happen. If you are using cut scenes to progress the story, describe the cut scenes in detail.

7. Game World – describe the general look and feel of world. Of particular importance is whether the world is broken up into different areas that are relevant to the game. If it is, describe how they are connected, how the player moves between them, and how they relate to levels.
8. Characters and Opponents
    - Player avatar. Indicate if the user has the ability to define their own avatar and what options they are able to define.
    - List of characters in the game: identify which are playable and which are not. If there are characters that are relevant to the game but never seen, be sure to include them. Include enemies and opponents. If the enemies are classes only, that is fine. Identify the class and if there is a specific number of them, etc.
    - For each character, give their backstory, personality, appearance and abilities
    - Explain the relevance of each player to the story and their relationship to other characters. If certain of the characters only appear in certain levels or areas, explain that as well.
    - This is also the place to describe any AI that is used for these characters.
9. Levels
    - General description of how levels are used. If they are very similar, the detailed level descriptions need not be very long.
    - For each level, describe what additional skills or information they are learning?
    - For each level, give a synopsis of the level and the objectives of that level. If there is introductory material that is required, how is it provided? There are two key aspects to cover in each level: the game play changes and the content changes. In some cases, both change; in other cases, only one does. For example, as levels progress, you may give the player less hints so that they learn the content better and you may increase the frequency of enemies.
    - For each level, describe how the map changes and whether it is a crafted or procedurally generated map. If it is procedurally generated, what are the constraints or requirements that put on it?
    - Is there a training level or a playable tutorial?
    - How is the player's progress assessed at the end of the level?
    - Are they allowed to replay levels? Does progression through the levels require a specific level of accomplishment (that is are subsequent levels being unlocked)?
10. User Interface
    - What is the visual system? Specifically,

      If there is a HUD, what is on it, how is it displayed and how is it accessed?
      What menus are included and how are they navigated?
      What is the camera model?
    - Control System – How does the game player control the game? What are the specific commands?
    - Audio, music, sound effects
    - Game Art – intended style
    - Help System
    - Settings – what can the user change?

11. Content Additions – How can the administrator add new content to the game? The information on the Functional Specification and Software Design Document pages may be useful in thinking about this non-game part of our project.
12. Feedback for the Player – How can the player tell if they are doing well? Is there mechanisms to tell them what they are doing wrong?
13. Data Collection for the Administrator – What information is being collected for the administrator? How can they access it?
14. Deployment – Describe everything that a person will need to run the game.
    - Where is it deployed?
    - How can they recreate it?
    - What information is needed to get it started? Do not, however, include an passwords in this document!
    - What systems are required to run the game?
    - If it is a game that runs on local devices, deployment includes how to rebuild it and where it needs to be stored for users to be able to download it
    - What configurations and settings are required to get the game working?
15. Development – What does a developer need to know in order to continue development on the game? The information on the Software Design Document pages may be useful in thinking about what needs to be captured.
    - What platform is it built on, including versions? Be sure to include all dependencies
    - Where are the source code and assets to be found? This should be in a publicly available repository.
    - What are the key elements of the game? This will differ considerably based on the platform. Where can a developer find details on the game structure? The details need not be included in this document. A reference is fine. Wherever the documentation is, however, it needs to be sufficient for them to find their way around the code and to understand what the different elements are. Be especially careful to identify the key problems that you encountered and solved. These are likely to be subtle points that the next developers will encounter as well.