

How to write an effectively communicating design document for games

How to write an effectively communicating design document for games

Examensrapport inlämnad av Jesper Bylund till Högskolan i Skövde, för Kandidatexamen (B.A.) vid Institutionen för kommunikation och information. Arbetet har handletts av Ola Holmdahl.

2008-05-07

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat _____

Jesper Bylund

How to write an effectively communicating design document for games
Jesper Bylund

Abstract

In this reflexive report I detail my process for creating effective documentation for computer game design.

I begin by explaining the background of my work, the goals and limitations I had during this thesis, my goal being to find good practices about writing effective design documents.

One chapter is dedicated to the methods I used while writing both the documentation, as found in appendix A, and this report.

I then document and discuss my work process and the choices I made as well as the basis of those choices, I also reflect on the effects of my changes.

My analysis is based on my iterative work, external sources and private experiences. Based on my work I conclude that creating documents that are easily searched and that only document the information crucial for implementation leads to effective abstract design documents.

Contents

- 1. Introduction.....1
 - 1.1 Background.....1
 - 1.1.1 Company: Lockpick Entertainment1
 - 1.1.2 Product: Dreamlords the Reawakening.....1
 - The convergence cooperation problem.....1
 - 1.2 Purpose2
 - 1.2.1 The idea.....2
 - 1.2.2 Problem definition2
 - 1.2.3 Limitations2
- 2. Method.....3
 - 2.1 Terminology3
- 3. Work process iterations4
 - 3.1 First iteration4
 - 3.2 Second iteration.....6
 - 3.3 Third iteration.....8
- 4. Analysis of my result10
 - 4.1 Macro versus Micro level communication10
 - 4.1.1 Macro level communication.....10
 - 4.1.2 Micro level communication10
 - 4.1.3 Conflict.....11
 - 4.1.4 Discussion: Macro versus Micro level communication.....11
 - 4.1.5 Rule of thumb: Macro versus Micro level communication11
 - 4.2 Redundant information.....11
 - 4.2.1 Discussion: Redundant information12
 - 4.2.2 Rules of thumb: Redundant information.....12
 - 4.3 Using illustrations12
 - 4.3.1 Discussion: Using illustrations12
 - 4.3.2 Rule of thumb: Using illustrations.....12
 - 4.4 Discussing design document use12
 - 4.4.1 Rule of thumb: Design document use.....13
 - 4.5 Discussing designer specific documents13
 - 4.5.1 Rule of thumb: Designer specific documents13
 - 4.6 Discussing the use of models at Lockpick Entertainment.....13
 - 4.6.1 Rule of thumb: Design document consistency14
- 5. Discussing my work process.....15

5.1 The design phase	15
5.1.1 Social problems and methods.....	15
5.1.2 Playing together.....	15
5.2 The documentation phase	15
5.2.1 Writing for the right people.....	16
5.2.2 A gender perspective	16
5.2.3 Deciding against established symbols	16
6. Conclusions	18
6.1 Concluding discussion.....	19
8. Referenses	20
From the Internet:.....	21

1. Introduction

My introduction to my work will lightly touch upon the background of my thesis and explain my purpose and the limitations for my report.

1.1 Background

The background details the foundation for my design and describes for what purpose the design was originally created. This serves as a context for my design and has no direct effect on my report on writing effective design documents but has effected how I have worked with my design document.

1.1.1 Company: Lockpick Entertainment

The design discussed in this report is based on a game by the Swedish game developer Lockpick Entertainment of which I was previously employed. The restrictions on my design by the quirks of the original game I can take no credit for.

1.1.2 Product: Dreamlords the Reawakening

Dreamlords the Reawakening (DLR) is the expansion of the original Dreamlords (DL) (*Dreamlords*, 2007 & *Dreamlords the Reawakening*, 2008).

Dreamlords the Reawakening is a massively multiplayer online game (*Alexander*, 2005), or MMOG, that combines short action focused real time tactical matches with an online management interface for long term strategic game play.

Players interact with the world of Dreamlords either through technology based progress which happened over long periods of time or through matches against AI controlled mobile enemies (mobs).

Players interact with other players through forums, chats, trades and player versus player matches. Players can and are encouraged to form teams of members to actively cooperate, these teams (called convergences) give members access to a private forum and an internal convergence trade.

Players are encouraged focus their progress in technology trees to specialize and become predefined roles, for example a crafter of items, player versus player expert or wood harvester.

MMOG's are based on the foundation that social interaction makes any activity more enticing and allows players to play for longer without growing bored (*Alexander*, 2005). But the rather limited interactions described above makes it hard, less rewarding or in some cases bluntly boring to cooperate with other players in DLR. Hence the need for a convergence cooperation systems design.

The convergence cooperation problem

At the release of DL there was a sense of direction for the development team that they would shortly release an expansion that enabled war to be fought between convergences. Wars that were fought in such ways that each member of a convergence needed to cooperate to win, regardless of their role.

Because of various market related reasons this expansion was never created nor even designed. The subject of my thesis is to create a design document that effectively communicates such an expansion for DLR.

1.2 Purpose

My main purpose in my thesis is to expand my own knowledge about how to structure and write design documents to effectively communicate abstract design for games.

The purpose of this report is to detail the process of my work, analyze and discuss the various solutions or problems that I encounter during my work.

I also intend to use my design document as a work sample in professional circumstances.

1.2.1 The idea

Design documents are used in software development as documentation for the functions and features intended to be implemented in a product.

Most designers have their own ways to communicate their intended designs because of the lack of standardization, but there has been an ongoing discussion about design documents in the games industry for years concerning how necessary they really are (*Jaffe, 2007*).

The idea behind my work is to iteratively achieve an effective document structure based on general good practices that can be used by other designers as well.

1.2.2 Problem definition

In my experience working in game development projects at the University of Skövde and as a game designer at Lockpick Entertainment there are problems with the effectiveness of design documents.

Developers tend to not read the documents or read one version and not the updates.

Designers use large amounts of time keeping documents up to date and contradictions in the design documents lead to problems with the development.

I intend to analyze how I can increase the effectiveness of design documents for my experienced context.

1.2.3 Limitations

I will base my evaluation of effectiveness on my experience working with design documents at the University of Skövde and at Lockpick Entertainment. All changes that I believe improves the readability and ease of use of a design document are making the document more effective. Changes that might impede the reader or the iterative development of the document are not making the document more effective.

My design documentation is meant to be used directly in the social and cultural contexts that exist at the University of Skövde and Lockpick Entertainment as I write this report.

More specifically it is tailored to be used by the current development team at Lockpick Entertainment.

The models and templates I use as foundations for my document have been selected on the basis of personal interest rather than scientific selection. These choices were severely limited as most professional game development companies do not release their design documents for public use.

2. Method

To begin I will first create a game design for documentation. I will choose a design documentation model and create a first design document using my design.

I will then compare this document to other design documents or design models to pinpoint the good and the bad practices with the model. I will then reiterate this process as many times as possible.

During my iterations I will document all the changes I am consciously making, what those changes are and why I make them.

I will then analyze the most interesting observations or changes I have made during my iterations. Through discussion I will try to locate good practices for game design documentation.

2.1 Terminology

Throughout this report I will use some words out of context or with a new meaning. At least one word I will use has no prior clear definition. For the course of this report I will therefore define the meaning of these words as I use them.

2.1.1 Fluff

Fluff is information that is meaningless to the reader I'm describing, at the time and in the role I describe them. Fluff might often be valuable information that is simply displayed in the wrong context or at the wrong time or for the wrong person. Fluff is very subjective and could be perceived as misinformation if it was not valuable in another context.

The most important aspect of fluff is that it is not currently useful information to the reader and therefore serves as a distraction.

Fluff is usually all the information that is not needed right now. It can be quite valuable information but can create just as much of a distraction as misinformation would. Fluff in design documents has many similarities with the idea of noise contaminating direct communication. Noise in direct communication is defined as outside disturbance that might cause the message from sender to receiver to deteriorate amidst competing data (Shannon, 1948).

2.1.2 Implementation

For the course of this document I use the term implementation, in many forms, to talk about all the development that creates a game. Programming, art and sound creation are all referred to as implementation.

I will often use the word as a generalization meaning the development of the feature being discussed regardless of the practical activities involved. I will for example refer to both writing code and creating graphics as implementation at the same time.

This is because the various acts of implementation have the same demands for information; they simply need the information for that act. My thesis intends to find a model of documentation that can cater to these different implementation acts at the same time.

2.1.3 Convergence(s)

A convergence (CV) is a grouping of players in DLR consisting of up to 15 members. The convergence has its own forum and trade system exclusive available to convergence members.

3. Work process iterations

I have described my iterations below detailing the conscious changes I've made during that time.

For each iteration I have rewritten the entire document based on new models or ideas as well as what I learned from the previous iteration.

3.1 First iteration

I began my first iteration by using a widespread template created by Chris Taylor. Taylor's template is formatted as a list of all features in the game as well as overview and pitch documentation for communicating overall vision and selling the game to publishers (*Taylor, 2008*). Taylor's model is available in appendix B.

Taylor's template seemed to cover all the necessary information about a game. It did not seem very effective as a tool because of its size, for a complete game I estimate at least over a hundred pages long.

My first observation about the template is that it presumed that the writer had absolute knowledge of the design. Its structure is simply a list of all features needed to create the game. It had no levels of abstraction from which a writer could organize the information.

It could also hardly be used until all the design was complete, as much information needed to be cross referenced as it was all described in the most fine grained abstraction level. From my experience this has been impossible to achieve as much of the development needs to be parallel to save time and money.

After a few days I took a step back from my document to find an amount of text hard to overview. It was too large for an individual to remember the entire document. While this is not strictly necessary for developers it also means that no designer can be sure the document does not contain misleading or outdated facts. For any member of the development team it did, however, contain large amounts of fluff.

I compared this version of the document to a model presented by David Jaffe (*Jaffe, 2007*) on his blog. Jaffe's example is the overview design document for the small electronically distributed game *Calling All Cars* (*Incog, 2008*) and is very different from Taylor's template. While Jaffe's overview document is probably used for different purposes than Taylor's template the structure and practices used can still be compared.

Jaffe's document shows one page for each game *state*. What a game state is he does not define but I interpret it as a portion of gameplay that could be implemented separately from the rest of the game, even though all states are based on the same graphical assets.

For each state Jaffe defines the choices a player can make and the results of the choices that affects the game mechanics. Jaffe's template is much shorter than Taylor's and also makes effective use of illustrations showing what the results described in the text means. When, for example, Jaffe describes how a player can toss a criminal into a paddy wagon there are two illustrations as well as the text description detailing the possible outcomes of the mechanic.

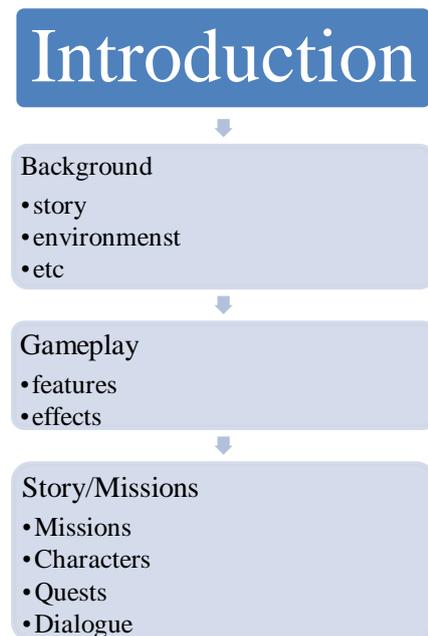


Figure 1.1 – my document based on Taylor's model. It was a mess of text.

In the end of my first iteration my document appeared very bloated and not very effective as a communication medium. Taylor's template was not effective for individual users because the readers were forced to read and memorize the entire document to get an overview. Much of this information would be fluff to any individual team member.

3.1.1 Reflections

Using Taylor's template will help make sure that designers don't miss crucial subjects to be addressed. But because of the sheer amount of information presented any developer will be confused by the contents.

Taylor's template also means that designers must have a clear image of all the details of the game before structuring the information in the template. This could mean designers would be using documentation in another form *before* they started using the template, making the template less useful for designers.

In an iterative development the document would have to be continuously updated and while it contains a section that is intended to document changes it would require several developers to re-read the sections changed with each new iteration.

Reading, or simply browsing, this volume of text takes time that might be much more effectively used. Especially since the designers writing the document would have the same problem as the intended readers in finding all the information and updating it. Longer texts in themselves require readers to put more effort into actually retrieving information which means that the longer the text is the less information it conveys for a set amount of perception. Writing a document that is long and cumbersome for the reader is not an effective way of communicating an intended design because it takes too much effort on the part of the reader. I rather feel that Taylor's template is aimed at the wrong demographic as it does not at all seem to reflect the needs, at least in my experience, of teams in stressful projects but rather the need of a designer to document everything about their design.

No matter how a product defining document is written another valuable lesson I have learned is that names are absolutely crucial. Using consistent names can be the difference between a clearly communicating document and fluff. But names always communicate connotations that might not be intended.

Giving something yet undefined, except for in the document, a graphical name can affect the implementation of that object or function in ways that might not be intended by the author. Changing names can also be hard as an entire group of people need to relearn what they already know. This type of relearning is usually met with lack of interest or if the name seems important the change might be resisted (*Sennet, 2007*).

I have one element in my design that I know with certainty will be renamed, for this reason I've given it a name that has nothing to with the game and sounds out of place and artificial (*patria verbs*, appendix A). Hopefully this separation will make the name transition easier.

The Calling all Cars design document shows some communication ingenuity on David Jaffe's part as he includes illustrations of the things described but simplified and redundant. Cartoon versions of the functions described that might help the reader to understand the feature being described quicker. This is interesting when compared to illustrations used to communicate additional information such as diagrams or flowcharts. Is the information presented in Jaffe's illustrations really redundant or do they minimize fluff for the reader by making the information more readily searchable?

3.2 Second iteration

My second iteration is still based on the feature segmentation found in Chris Taylor's template but I try to use David Jaffe's overview technique in how I structure the information (see figure 1.2).

The document begins with an overview section describing the different features and how they interact to give the reader a high level understanding of the system. I then add a player experience section that describes how a player can use and feel about the features described. This section is written as a story of a player's experience with the game. The section is meant to communicate soft values and a meaning of the design to developers. These sections are then followed by a feature by feature technical description much like that of Taylor's template.

This disposition of information is easier to search for needed information which is suitable for the diverse demographic a design document is written for. Graphical artists and programmers rarely need the same information any documents that can be used by both groups save a lot of time.

I wrote the descriptions of features and overview paragraphs in short and simple denotative language (Russell, 1956). But this format for the feature texts still produce fluff, as readers must read longer texts about the feature to find the intended information.

All this extraneous reading might communicate information that is needed elsewhere in the project but it also puts some strain on the reader's perception. According to Mann's principle of least effort any reader will simply chose the most accessible way to search for information and then stop searching when any viable information surfaces (Mann, 1993).

The problem being that my documentation does not let the reader search for information without reading long paragraphs of text. Trying to find a way around this problem I again look at Jaffe's design document and realize that all his text is written in short paragraphs of about one to three sentences long. These paragraphs deal with one small part of a feature each.

Based on Jaffe's short paragraphs I restructure my design document accordingly. My document still has an introductory overview but features are now represented in short paragraphs each containing vital information making information searching simple.

At the end of my second iteration my text seems fractured and still hard to overview. I believe I am trying to convey more than the information needed.

3.2.1 Reflections

The disposition of overview section and feature section worked very well to give an immediate overview. An unplanned side effect of this overview is that the macro level information makes information search in the design much quicker and simpler.

The thick paragraphs of text create an enormous amount of fluff for readers while searching for information. As my design document is intended to be used in production it is supposed to be constantly referenced and updated during development. Searching through fluff will lead many to abandon searching the document and instead directly ask the designers which is counterproductive to the parallel work flow.

Extrapolating on Mann's principle of least effort I believe that communication has only a short window of time in which to take place before the reader's attention wanders. To minimize the time the reader needs to focus their attention, information searching is very

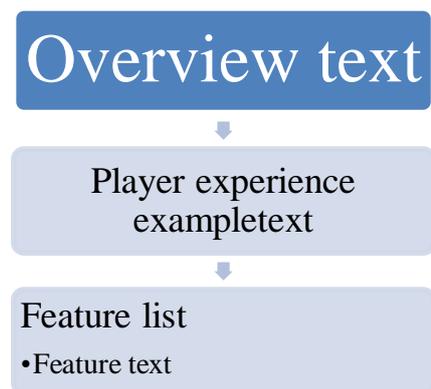


Figure 1.2 – the second iteration documentation model

important to design documents. Bullet lists and pictures therefore should be more effective and communicative, conveying *more* information in a *less* data dense form than the more data dense texts can.

I find that this model for documentation is aimed at the wrong demographics. It does not convey the needed information with the speed and clarity that is needed. The urge to write long paragraphs of text might come from a perceived need to communicate *all* the information about a design, when in fact the designer should be aiming to communicate the *necessary* information.

To be most effective this model forces the player to read the introductory overview at least once. While reading the overview will help most readers find information more quickly it is not ideal to force readers to read any longer texts as this contradicts the principle of least effort.

To create an effective design document that does not force readers to filter between fluff and information I found the need for an analytical system that defines the atomized features of the game. I need to divide all features that might increase effectiveness while making sure I do not divide unnecessary features. To make these choices the analytical system must help me view the game as an instanced system so that I can define what features can or cannot be divided further.

For the documentation to be structured effectively it needs to be searchable for the end user, in this case the game development team, and not written for the designers themselves.

3.3 Third iteration

My third iteration is heavily based on Jaffe's design document. The main layout of the document is the same as in iteration 2. The documentation still begins with an introductory overview detailed in iteration 2, followed by a player experience section also detailed in iteration 2. I updated the feature by feature walkthrough based on my observations on information searching. Each feature is now divided into two sections; the first being a feature header which is an overview a few paragraphs long. The second section is a feature walkthrough detailing the features atomized parts using bullet lists and very short text paragraphs.

My design is not suitable for redundancy illustrations because it is mostly based on menus and I have only found use for such images in two places. Redundancy illustrations can easily show objects and changes to that object, movement, direction. It is less effective to use illustrations that do not show these types of changes, for example item change, menu states. Therefore I find that the function of redundancy illustrations is closely linked to visual changes (game changes that are perceived by the player visually. Example: the players game character jumping). Because my design has very few visual changes I have chosen other ways of detailing information that I believe to be more effective.

I have used flowcharts throughout the document to detail non-visual change based information that is hard to convey using text. Flowcharts are not redundant but simplify the information being communicated because the hierarchy that needs large amounts of text to be explained is inherent in the structure of the chart.

Wherever possible I have also used bullet lists to describe what happens as the players make choices.

This design document model is far removed from Chris Taylor's template. It is similar in many ways to David Jaffe's design document but is not structured in the same way. My document is structured after features while Jaffe's was structured after game states.

To isolate what features that could not be efficiently divided into sub features I used an analytical model created by Ben Cousins. The model is intended to be used to analyze how game mechanics work but I've been using it as a way to clarify my design. Cousin's model is based on four layers of increasingly finer details of the game experience. The first level spans the entire game experience; the second spans each sub goal for the player, the third tactical decisions by the player. The finest detail level, the fourth, being the atom of games mechanics and s defined as the smallest input a player can make that effects the game. (Cousins, 2005) I used this model to identify what features I could divide until I reached Cousins third detail level. Because my design is completely based in graphical user interface the only lower level input players can make is mouse clicks which I deemed unimportant to detail in my documentation.

Cousins' model also makes it simple to realize what information should be conveyed to developers as extraneous information is not present in his lower detail levels. I can then judge whether this information is fluff or not and use it in overview texts.

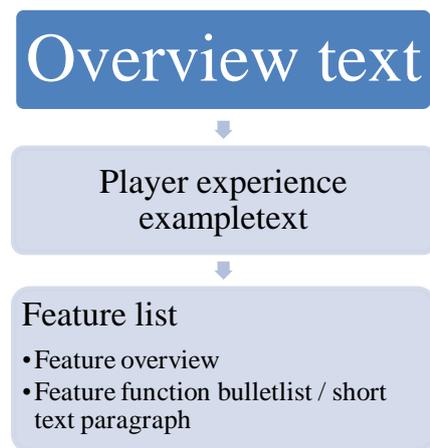


Figure 1.3 – the third iteration documentation model

3.3.1 Reflections

This iteration is based on Ben Cousin's analysis model. While atomizing features was not a viable strategy for my design, for the reasons I have described in [iteration 3](#). The model did however teach me to discover an appropriate abstraction level for the document.

Using Cousins' model I divided features into atomized effects of player inputs and then recreated the features into the level of abstraction that showed what the feature was intended to do. With the intention clear I could then write very simple text to detail the feature. This method also separated most of the fluff information that designers unnecessarily try to convey in the documents I have read.

I structured the document for easy navigation and information searching as my previous iterations showed that the lack of searchable information produced large amount of fluff. The structure to improve information search is in part borrowed from Jaffe's design document but also from the structure of Taylor's template as headers inside individual features can increase overview of that feature and make the short paragraphs of texts easier to organize.

While this iteration of my design document conveys a simplified design it effectively communicates the crucial details and an overview of intent. Crucial details and overview of intent work overlapping each other to communicate a clear picture of the design with very little fluff. So while it does not contain even close to the amounts of data of Taylor's template I believe it communicates more information and in a more reader friendly way, thus making the model more effective.

4. Analysis of my result

This analysis of my result and the discussion surrounding it are used as the basis for my conclusions.

4.1 Macro versus Micro level communication

Macro is a word derived from the Greek word *makro*, meaning long or large, it is often used as to describe viewing something of a very high level of abstraction (Nationalencycledien², 2008).

Micro is a word derived from the Greek word *mikro*, meaning small, it is often used to describe viewing something on a very close level of detail (Nationalencycledien¹, 2008). During the writing of my design document I have noticed that conflict between macro and micro level information leads to large amounts of fluff.

4.1.1 Macro level communication

All the people connected to game development need to understand the end vision of the product in order to effectively work together against a common goal.

This means that macro design communication is extremely important and needs to be done continuously as production goes on, both as a reminder and update to developers when changes are made on the project.

Macro design is often seen as hard to communicate because it is most often the sum of the many parts the game consists of. But this view of the macro level design does not actually focus on the vision of the game but a definition of the game.

The vision of the game can be simplified without losing information. Mario for example can be defined as; *joyful jumping in a colorful world, getting passed grumpy enemies*. The description does not need to define *what* is being jumped on or *how* a player stomps enemies because it still communicates the central vision.

This form of macro level can be communicated through illustrations and does not need a lot of words except to capture the function, as the illustrations capture the feelings. For example a picture of a happy Mario in the air over an enemy is more informative when the word jumping is written next to it.

Macro level design that needs to communicate an accurate model of how features function together is hard. From my work I've come to believe that design documentation needs to do this two-fold: one text that in detail lists the features and how they are interconnected (e.g. Mario can *jump*), but also a text that shows the readers how a player will use the features, for which I prefer a short player interaction story. This communication is important because the reason behind a feature can be much more informative than details on how it works.

4.1.2 Micro level communication

The main use of my design documentation is to guide the implementation of the game in order to reach the development team's predefined goal. Micro level documentation of features performs this function as designers can plan out gameplay in advance and then give the implementation team instructions on what they should implement and how it should work.

Micro level information is usually very technical and not at all suited to be read by all members of the team. A graphical artist has little use of the mathematical algorithms that create a jump in a Mario game. The person implementing the feature needs to know, someone else might perhaps use the documentation to see how the feature works but the audience is very limited.

Micro level information is best shown as algorithmic structures because they are the most similar information to the implementation level possible. One could argue that a

documentation written in a programming language might be closer, but the interpretation that needs to be done by programmers is roughly the same. This similarity removes most of the fluff from the text and lets the person working on the feature use the documentation without much interpretation. It is however not reader friendly for achieving an overview of the feature, which is essential before implementation begins as it often affects the implementation style.

4.1.3 Conflict

These two levels of communication are in direct conflict as they are mostly fluff respectively to each other.

Separating the two with white space does not make the distinction clear enough because it can just as easily be interpreted as ending and beginning another feature or part of feature. Using paragraphs of normal text for macro level and bullet lists for micro level information has been a successful approach in my design because it clearly separates the two and is a simple model for the readers to learn.

The most important thing in order to limit the amount of fluff is to clearly separate the two abstraction levels so that no one will ever read the wrong text when looking for only one level of information. This improves the effectiveness of search which limits the distracting effects of fluff.

4.1.4 Discussion: Macro versus Micro level communication

Macro and micro views of a design exist because of the needs of the developers who read the design documentation. Exactly what these needs are can, and with certainty will, change depending on context of the developers. An entry level programmer with no power over the production in a Japanese development team of 500 members will probably not have the same demands on documentation as a western graphical artist in a development team of 5 members. The conflicting nature of micro and macro views of design documentation is arguably based on my definition of the micro and macro level views. But overviews and detailed views are needed to effectively communicate an intended design.

The argument that everything could be learned from the details could be made. But it is essentially flawed as the purpose of the documentation is not to give a complete and accurate view to every developer but to communicate the information a developer wants as fast and accurately as possible.

4.1.5 Rule of thumb: Macro versus Micro level communication

- Use macro level overview clearly separated from micro level details, the more separate the better. If much macro level information is needed place it in a separate document.

4.2 Redundant information

Redundant information is not usually functional in documentation. It may be distracting for the reader or in best cases simply tiring for a reader's perception. I have therefore classed most redundant information as fluff throughout this report. But in some cases redundant descriptions using a different medium or point of view can be used to make sure the reader understands a specific point.

Illustrations and flowchart can communicate information that would be ineffective through text such as hierarchies because they can simply show the reader the hierarchy. In the same

way illustrations and flowcharts can make the points of a text obvious to the reader by showing a simplified model of the information or an example of the feature. Redundant information in the form of illustrations and flowcharts can therefore improve information searching and information communication in design documentation.

4.2.1 Discussion: Redundant information

There is always the chance that illustrations and flowchart are misinterpreted and therefore acts as misinformation. This is always a problem for the author when redundant information is available. But as it is a problem based on the quality of the author it does not negate the good practice of using illustrations and flowcharts as redundant sources of information. This problem is further discussed in [4.3 Using illustrations](#).

4.2.2 Rules of thumb: Redundant information

- Minimize the reader's exposure to unnecessary information. That includes crucial information that is not necessary at the time.

4.3 Using illustrations

Using illustrations for communication can be ineffective because of the interpretive nature of illustrations. If you can be sure that no one reading your document can misinterpret an illustration then using it will be a fast and in many cases ideal communication method. From my experience and the example of Jaffe's design document illustrations can almost always be misinterpreted by the reader. This does not mean that illustrations are not effective for communication, but it does mean that they cannot alone communicate their information without the risk of becoming misinformation.

Using illustrations together with short descriptions of what they depict will allow most readers to understand the illustrations intention and thus interpret it in the intended way.

4.3.1 Discussion: Using illustrations

Illustrations are communicative tools that I have not focused on during my thesis. Illustration is simply too large a subject to include in my work.

I have in my comparisons noticed that my own illustrations need *more* descriptive text than my professional counterparts. This leads me to believe that the quality of illustration is oppositely proportionate to the need for descriptive text. My own illustrations of comparatively poor quality need more descriptive text than the illustrations of professional designers. Some text is however always needed in my experience.

I therefore believe that collaborating with an illustrator can be a very good idea for most designers.

4.3.2 Rule of thumb: Using illustrations

- Don't leave illustrations as the sole information source. Use them as a layer of redundancy for important information.

4.4 Discussing design document use

While reading design documents online I realized that another possible function of a design document is to allow developers to follow up the implementation.

Even with perfect documentation of a design, the implementation will define the feature in a way that might not be suitable for the features intended result. Design documents can be used by designers as a way to measure the implementations validity against the intent of the design. This use of a design document is another reason to make any design document easily searched for information.

It can also work as a reality check for designers. Is this how others interpret the meaning of this design? Is this interpretation better suited for our goals than my own version? These are questions that might help a development team progress towards better design and implementation cooperation.

4.4.1 Rule of thumb: Design document use

- Make design document easily searchable and readily available. If a developer needs more than one minute to update the document, chances are that the task will be less attractive and therefore happen less frequently than if the documents were easier to update.

4.5 Discussing designer specific documents

While writing my own design document and reading documents created by others I've noticed a certain trend to write about why a feature exists or why a part of a feature is designed that way. This information is important to persuade developers to follow a design and not distrust systems just because they might not understand the intended result.

But writing about the intended use of the feature or describing why it will work and be fun is mostly redundant and fluff in a design document, especially when this is added to every feature.

The information itself is very important but defending design is not an effective use of a designer's time (though in some cases absolutely necessary). Because of this I recommend creating a separate document that details the intention, effect and purpose of each feature. This document can be used as a reference for designers when asked about features designed very early or it can be used to bring designers up to speed on current systems without forcing them to learn the minute technical details.

4.5.1 Rule of thumb: Designer specific documents

- Keep designers informed with separate documents to avoid conflicting information at the implementation level.

4.6 Discussing the use of models at Lockpick Entertainment

During my time at Lockpick Entertainment no models for design documents were used. This can be an advantage for designers as it allows them to be flexible when documenting abstract design. But it can also be a serious drawback for the development team.

When members of the development team search for specific information the different layouts of data will make it harder for them to find the information they need. This can lead to misinformation and, in my experience, a tendency for developers to avoid reading the design documents.

When the design documents are no longer the easiest way to find information I have observed that sometimes developers start asking designers directly instead of reading the documents, which negates the function of the documentation. It also makes the designer's job much harder as they will be constantly interrupted. The developers will also be less able to work on parallel tasks as the information will be limited to their access to the designer.

In some cases I believe this downward spiral might lead to developers procrastinating when they've finished a task or possibly doing unplanned tasks as they feel that it might be hard or boring to start reviewing the documents needed for the next planned task.

I have observed developers in this downward spiral during projects at the University of Skövde and when this basic trust for documentation fails it has, in my experience, been close to impossible to regain.

Using my model as a basis for design documents would make searching for information much easier as the data would be consistently structured. But the entire development team would need to learn how this model is structured as well as the designers learning to document according to the model and sticking to it as they work on a day to day basis.

Changing documentation model to a model such as mine during an ongoing project can take time and effort on behalf of the developers. But it might also save a lot of time if their previous model is less effective.

I believe that using a model is efficient for any development team. But whether they should use my model or some other design documentation model I cannot say. Additional research might be required, or different teams might prefer different models. Standardization for abstract design would lead to a massive improvement in efficiency as the search and research times would decrease. But a standard that is effective and general enough is probably years away, I believe that game developers or academics first need to find the basic building blocks for games before a documentation model can be truly efficient.

4.6.1 Rule of thumb: Design document consistency

- Use consistent models for communicating design documents. This improves the development team's ability to search for the needed information.

5. Discussing my work process

During my entire writing process I gathered many thoughts and reflections about my work process. I will now discuss the thoughts and reflections that I have deemed most important or interesting.

5.1 The design phase

The design phase of my thesis was solely dedicated to creating the design I would later document. It was a short period but it coincided with my research about writing for groups and the mix led me to some interesting thoughts.

5.1.1 Social problems and methods

Reading about the changes in society that the new economy brings into action I realized that one of the hardest problems for multiplayer games to overcome is the alienation that naturally occurs between differently skilled players.

Games must of course allow players with different skill levels to play but what I mean by alienation is the segregation that happens socially between players when skill levels vary too much. Varying skill levels often means that the better player is waiting for the less proficient player or that the less proficient player finds it utterly boring to play with the better player. The problem is a parallel to the problem of difficulty in games; if it is too easy or too hard it will very quickly become boring (*Koster, 2005*). But alienation based on skill difference can keep players from playing the game at all, if the game is focused on the social gaming demographic.

Singstar (Sony Computer Entertainment Europe, 2004) and *Guitar Hero (Harmonix, 2006)* are excellent examples of games that don't have this problem or at least do not obviously suffer from it. But how many other multiplayer games may suffer from this?

One solution is to create game systems that allow for asynchronous difficulty levels in multiplayer. But many such systems can be seen as handicap for less skilled players if they are too obvious, making the system irrelevant to the problem.

5.1.2 Playing together

People around us affect our emotions. Just as we may, by forcing ourselves to smile, make us more positive or even happier (*Eysenck and Keane, 2005*). Transferred to games this means that fun can be more fun if other people are having fun with the game at the same time. Their happiness will "rub off" on the other players.

This effect is not always correctly interpreted if it is at all used in games. Most multiplayer games are based on conflict between players, why this is I have not tried to analyze or define but in my own experience it is easier and less costly.

Because joy spreads between people they must experience joy at nearly the same situations, not in opposite situations as is multiplayer games based on player conflict.

Cooperation is therefore very important while working with transient joy. Creating an MMOG using player classes that cooperate is a tried and functioning way of using transient joy but as long as the game itself is based on conflict there will be opposing players that are excluded.

I did not intend to solve this issue but thought it valuable to detail in preparation for my discussion about my documentation.

5.2 The documentation phase

I wrote several iterations of documentation, based on models and templates, to find good and bad ways of documenting interactive design. My decisions about what models to use and why

was partly explained in the iterations chapter. Here I intend to detail some of the things I decided not to use and some of the problems I did not find solutions to.

5.2.1 Writing for the right people

One aspect of design documentation I noticed traces of while reading others design documents is designers are writing for the wrong audience. I then stumbled upon the same error as I wrote my first iteration.

Design documents are often held to be the only documentation of the game in development. Even if the documentation is separated into many documents this is still true in my experience.

As designers document the design they seem to want to document everything about the design. This has been a problem in many of the documents I have read and written as out of context information is placed randomly throughout the document seemingly as the thoughts occur to the designers.

But the purpose of a design document is to communicate a design to the implementation staff. All the information that I found in my first iteration and in other design documents was interesting and informative about the game but completely irrelevant in the context I found it in.

I believe that there is a problem with designers documenting for others before they document for themselves. A possible solution would be to have separate documentation primarily for designers.

The counter argument that the designs I refer to are simply poorly structured might also be true. But because this is such a common event, and because much of the information about a game (tidbits of culture, why a using a certain feature is fun, the default values for systems) has no real relevance for implementation I believe that documentation for designers is simply inadequate and should be separate from the documentation meant for documentation.

5.2.2 A gender perspective

All of my sources for design documentation are written by men. Because my sources are used as the basis of my documentation I believe that this affects the cultural perspective of my documentation. But I do not think that my documentations effectiveness is limited by my sources.

Before discussing this issue I had to check whether all my sources were male or not. This makes me wonder if there is any gender specificity in documentation but it is not a question I will even try to answer.

Because my documentation is focused on effectively communicating information to different special interest groups it should be relevant to members of these groups regardless of sex. Therefore, for the purpose of my thesis, I very much doubt that gender has any effect on the effectiveness of my documentation.

5.2.3 Deciding against established symbols

Using symbols was one of the first things I believed could change the way abstract design could be documented. Symbols are used in our daily lives to communicate meaning where words would take too much time or space (e.g. symbols used for men's or women's restrooms).

This lead me to believe that symbols communicate on a different level then words, that they are less prone to be misinterpreted because they are more exactly defined. This turned out to be wrong.

While researching symbols I learned that they are interpreted differently by readers (or viewers perhaps) much like normal texts or words. The difference being that symbols are

general enough that different interpretations give the same perceived effect. (Gudykunst and Young, 2003)

The symbol of a man on a door might for person A be interpreted as “only men allowed” while person B interprets the symbol as “men’s room”. The result is the same but the symbol has communicated different meanings.

For this reason I believe that symbols are hard to use for communicating abstract design. Symbols can be used in or together with illustrations with little effect of this problem but I cannot use them as a communicative medium in themselves as I had intended.

6. Conclusions

My conclusions are based on my analysis, reflections and discussions that I have documented earlier in this report.

The most effective and simple way of improving a design document is by writing well. Short denotative language that is too the point and never confusing is much more informative then longer more cumbersome texts that might contain more data.

Shorter paragraphs are easier to search for relevant information; the shorter the paragraph the better. Longer texts can hide important information in fluff. Make texts as short as possible.

Preferably use lists to show information already structured, leaving the structure out of the text frees up a lot of space and removes a lot of fluff.

Illustrations are fantastic at communicating soft information about a feature. Especially visual change can be shown with great accuracy and ease using illustrations. But never use illustrations alone, always use them as a redundant source of information writing it in text nearby.

Diagrams and flowchart are excellent vessels of information just like illustrations, but diagrams and flowcharts do not always need redundancy.

Using an analytical structure for both the design and the design documentation makes structures more informed and less error prone. I recommend Ben Cousins' analytical model to any designer.

Using specialized documents for specific tasks makes information updating and referencing much easier. Using additional specialized documents intended for designers can also simplify the structure of the rest of the documents.

6.1 Concluding discussion

My purpose for this thesis was to determine good practices for increasing the effectiveness of game design documents within the limitations of my experience. I would do so using a method of iterating design documentation while comparing my results to external models and documents.

For this purpose I have found a series of good practices that have increased my perceived effectiveness of design documentation.

It could be argued that my limited experience of development makes this thesis less stable than would be preferred for the subject. But because of my constant comparisons with more established or at least professionally used design documents I believe my arguments are very solid. Because my model is based on working effective design documents it is not limited by my experience.

My measurement for effectiveness is based on removing obstacles that impede the reader in finding the right information when needed. Had I analyzed this question further from the start I would have saved a lot of work as it is obvious that I am looking for a model that lets developers search for information quickly in design documents.

Jaffe uses a model similar to mine which lends credibility to my model as both seem created for searchable documents with a clear overview structure. If I had been able to acquire more professionally used design documents my conclusions could have been better defined.

My discussion can be perceived as loose or too quick. This is because the subject I chose has no currently supporting research and though I believe that my arguments are well founded further research might prove otherwise. My arguments are however based on real development experience and comparative research that is well founded.

My best argument for my conclusions is really the documentation that I have written, reading my design document (included as appendix A) and comparing it to Taylor's model (*Taylor*, 2008) or Jaffe's design document (*Jaffe*, 2007) clearly shows that my conclusions are based on the most effective features found in their work.

8. References

8.1 Books

Alexander, Thor. (2005) *Massively Multiplayer Game Development 2*, Charles River Media inc, ISBN 1-58450-390-4

Eysenck M W, Keane M. (2005) *Cognitive psychology: a student's handbook*, Psychology Press Ltd, ISBN 1-84169-359-6

Gudykunst, William B and Young, Yun Kim. (2003) *Communicating with Strangers*, McGraw Hill, ISBN 0-07-232124-5

Koster, Raph. (2005) *A Theory of Fun*, Paraglyph Press, ISBN 1-932111-97-2

Russell, Bertrand. (1956) *Logic and Knowledge*, Routledge, ISBN 0-415-09074-1

Sennet, Richard. (2007) *Den Nya Kapitalismens Kultur*, Bokförlaget Atlas, ISBN 978-91-7389-307-7

Thomas Mann. (1993) *Library Research Models: A Guide to Classification, Cataloging, and Computers*, Oxford University Press US, ISBN 019509395X

8.2 Games

Harmonix. *Guitar Hero*. RedOctane. 2006, (PS2)

Incog Inc. Entertainment. *Calling All Cars*. Sony Computer Entertainment America. 2008, (PS3)

Lockpick Entertainment. *Dreamlords*. Lockpick Entertainment. 2007, (PC)

Lockpick Entertainment. *Dreamlords the Reawakening*. Lockpick Entertainment. 2008, (PC)

Sony Computer Entertainment Europe. *Singstar*. Sony Computer Entertainment. 2004, (PS2)

From the Internet:

8.3 Articles

Cousins, Ben. (2005) Measurement Techniques for Game Designers *Gamasutra*

Retrieved April 28, 2008

http://www.gamasutra.com/features/20050512/cousins_01.shtml

Shannon, Claude. (1948) A Mathematical Theory of Communication *Bell labs*

Retrieved April 28, 2008

<http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html>

8.4 Blogs

Jaffe, David. (2007) Calling all Cars Design Document *David Jaffe's Blog*.

Retrieved April 14, 2008

<http://criminalcrackdown.blogspot.com/2007/02/calling-all-cars-game-design-document.html>

8.5 Encyclopedias

Nationalencyklopedin¹

Retrieved May 6, 2008

http://ne.se/jsp/search/article.jsp?i_art_id=255772

Nationalencyklopedin²

Retrieved May 6, 2008

http://ne.se/jsp/search/article.jsp?i_art_id=249098

8.6 Game Design Document Templates

Taylor, Chris. (N/A) Design document template.

Retrieved April 14, 2008

<http://www.designersnotebook.com/ctaylordesign.zip>