



**DEPARTMENT OF INFORMATICS**  
TECHNICAL UNIVERSITY OF MUNICH

*Master's Thesis in Informatics: Games Engineering*

# **A Practical Approach to Documents in Game Development**

**Alexandru Constantin Balotescu**







**DEPARTMENT OF INFORMATICS**  
TECHNICAL UNIVERSITY OF MUNICH

*Master's Thesis in Informatics: Games Engineering*

**A Practical Approach to Documents in Game Development**

**Ein praktischer Ansatz für Dokumente in der  
Spieleentwicklung**

<b>Author:</b>	<b>Alexandru Constantin Balotescu</b>
<b>Supervisor:</b>	<b>Prof. Gudrun Klinker, Ph.D.</b>
<b>Advisor:</b>	<b>Daniel Dyrda, M.Sc.</b>
<b>Submission Date:</b>	<b>May 15th 2023</b>





I hereby confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, May 15th 2023

Alexandru Constantin Balotescu



*“Documentation is a love letter that you write to your future self.”*

*— Damian Conway, book author & computer scientist*





## Abstract

This paper aims at facilitating documentation writing in game development by creating a volatile and customizable system that helps developers navigate the process of creating a game. To this end, the thesis proposes the *Ludodome* framework – a comprehensive mechanism designed to address complex challenges in the documentation process of modern game development. With its structured approach and systematic methodology, the *Ludodome* framework provides a practical and effective way to analyze, document and plan the transition from a simple game idea into a full-fledged implementation, with an emphasis on pre-production, but also with various beneficial implications further down the line, into production. It incorporates a set of principles and guidelines that help steer decision-making and ensure design consistency. In essence, the *Ludodome* guides the developers from the primal game idea – the *concept*, through pre-production, where long-standing, robust game *pillars* are established in order to conserve the game’s guiding principles throughout the whole development process. It then sets the stage for the production phase by paving the way for creating coherent, well-founded *user stories* via *features*, all while striving to achieve the final goal of the game. Finally, the resulting artifacts provide a unified collection of game development documents.



# Table of contents

<b>Abstract</b>	<b>9</b>
<b>Table of contents</b>	<b>11</b>
<b>1. Introduction</b>	<b>13</b>
1.1. Problem Statement	13
1.2. Outline	14
<b>2. Related Work &amp; Resources</b>	<b>15</b>
<b>3. The purpose of documentation</b>	<b>16</b>
3.1. Benefits	16
3.2. Who is it for?	16
<b>4. The Ludodome</b>	<b>17</b>
4.1. Etymology	17
4.2. Introduction	17
4.3. Concept, Pillars, Features & Goals	18
4.4. Concept	19
4.4.1. Definition & Goals	19
4.4.2. Examples	20
4.5. Pillars	22
4.5.1. Definition & Goals	22
4.5.2. Textual Pillars	23
4.5.3. Visual Pillars	25
4.5.4. Acoustic Pillars	31
4.5.5. Custom Pillars	33
4.5.6. Unique Pillars	34
4.6. Features	36
4.6.1. Definition & Goals	36
4.6.2. Example	37
4.6.3. Features vs. Pillars	40
4.7. Goals	43
4.7.1. Definition & Goals	43
4.7.2. Examples	44
4.8. The Ludodome in practice	47
4.8.1. Alignment table	48

4.8.2. Timeline	52
<b>5. Conclusion</b>	<b>54</b>
<b>6. Future Work</b>	<b>55</b>
<b>Ludography</b>	<b>56</b>
<b>List of Figures</b>	<b>58</b>
<b>List of Tables</b>	<b>59</b>
<b>References</b>	<b>60</b>

# 1. Introduction

## 1.1. Problem Statement

It seems that throughout the whole industry, writing documentation is a headache for most game developers. A proper game documentation clearly requires a lot of extra time and effort – it covers not only the design, but also the story, art, audio, technical details and many other aspects of a game. But the reason why so many designers dread it comes from a psychological point of view: because they see it more like a necessity rather than something people actually enjoy doing.

While some people (e.g. Baldwin<sup>[1]</sup>) have created extensive, concrete templates for game design documents, others argue that such *magic templates* are just a myth (Schell<sup>[2]</sup>) and cannot possibly work for all sorts of games. While I agree that the one-size-fits-all approach may not be ideal for game design, I also believe in finding a conceptual framework that facilitates the creation of documentation for any kind of game. Unlike a *fixed template*, a framework offers a set of guidelines that developers may customize and use as they see fit, depending on their project's needs – similar to how the *agile manifesto*<sup>[3]</sup> works. Moreover, frameworks allow for a more granular control<sup>[4]</sup> within its bounds, compared to templates, which are more restrictive.

This is precisely what this thesis aims to achieve: creating such a system that allows huge volatility in writing documentation, all within the easy-to-follow guidelines of a framework. The thesis also goes to show how documentation can be transformed from *just a necessity* into a powerful tool that guides developers throughout their project and keeps all departments of the game studio aligned.

## 1.2. Outline

This thesis is structured in 6 chapters. Chapter 1 presents the problem statement that motivates the decision of designing a framework for this thesis, rather than a template. Chapter 2 discusses related work and the resources used throughout the thesis while Chapter 3 briefly details what benefits documentation brings in general and who is its target audience. This is then followed by a rather big section, Chapter 4, where the thesis' main part is featured, the *Ludodome*. Here, individual components of the framework are discussed in detail, accompanied by numerous examples, some of which are the result of extensive research, while others have been created as part of the thesis. Chapter 5 is where all the conclusions are drawn, and lastly, Chapter 6 contains suggestions for future work.

## 2. Related Work & Resources

The conceptual idea of the Ludodome was first sparked by D. Dyrda's presentation on "Game Design Methods and Tools: Documents in Game Development" as part of the "Introduction to Audio for Games" lecture<sup>[5]</sup>. He presented the idea of bringing pillars, features and goals together into a unified construct as part of a recipe of how to create game design documents. From there on, I have started researching each of these components one by one, going through many different sources. While there is quite extensive research on all of these components, the different definitions I found in papers, articles and books alike varied a bit too much from each other in ways that would not allow them to be put together, so eventually I decided to define them myself in such a manner that they work seamlessly together.

Further notable inspiration resources include, but are not limited to:

- J. Schell's "The Art of Game Design" (Chapter 5)<sup>[2]</sup> for the concept;
- A. Andersen's article "Game Audio Guide"<sup>[6]</sup> for pillars;
- M. Eff's article on "Pillars, Goals, Features & Elements"<sup>[7]</sup> (which was coincidentally written on January 4th, after this thesis had already started) for both pillars and features;
- D. Mullich's article "Game Mechanics Vs. Game Features"<sup>[8]</sup> for features;
- D. Dyrda's lecture presentation for goals<sup>[5]</sup>;
- A. Dotsenko article on "Game Design Framework"<sup>[9]</sup> for the alignment table;

## 3. The purpose of documentation

### 3.1. Benefits

Documentation plays a very important role in software development in general, and in game development even more so. Still, only 4% claim that they always document their processes, as per this trend report by Paul Harmon<sup>[10]</sup>. Without delving too much on this topic, as it is slightly out of scope for this thesis, it is important to emphasize what benefits clean documentation brings with it. The following list has been put together by Atlassian<sup>[11]</sup>, the company behind Confluence<sup>[a]</sup>:

- A single source of truth saves time and energy
- Documentation encourages knowledge sharing
- Documentation cuts down duplicative work
- It helps with onboarding new team members

### 3.2. Who is it for?

In the past, writing a comprehensive documentation was crucial for both players and developers alike, as it was not only used to provide guidance for the developers during development, but also to teach the game to the players. This often resulted in huge user manuals, some nearing even 100 pages, like the StarCraft<sup>[b]</sup> manual from 1998<sup>[12]</sup>. Nowadays, user manuals have been replaced by comprehensive tutorials or various other web-based resources. Instead, manuals have become more of a luxury now, being sold with special edition releases.

One of the goals of this thesis is to create a documentation methodology for developers. “From developers for developers” is the driving motto. However, thanks to all of the different resources it generates, the resulting artifacts of the Ludodome may be used both internally as documentation and externally as a game manual or artbook.



## 4. The Ludodome

### 4.1. Etymology

The term Ludodome [ˈluːˌdooˌdooɪm]<sup>1</sup> has been specifically created for this thesis and is composed from the Latin words *ludus* (“game”) and *domus* (“house, home”). The word “game” was chosen to highlight the purpose for which this methodology has been created – game development, while the word “house” works as a mnemonic<sup>2</sup> to help remember the general structure of the framework (depicted in the next section 4.3 *Concept, Pillars, Features & Goals*), resulting in the composed term “house of games”. The metaphoric parallelism to the phrase “house of cards” is intentional, so as to underline how important it is for the individual components of the Ludodome to be robustly established in order for the framework to be able to deliver a successful outcome.

### 4.2. Introduction

The Ludodome fuses a variety of different terminologies from the game development together into a cohesive composition. Design concepts, pillars, features and goals have all been redefined to fit together in a meaningful way within this framework.

---

<sup>1</sup> Phonetic transcription by the International Phonetic Alphabet standards

<sup>2</sup> a device such as a pattern of letters, ideas, or associations that assists in remembering something

### 4.3. Concept, Pillars, Features & Goals

The following diagram helps envision the seamless concatenation of concept, pillars, feature and goals into a comprehensive construct:

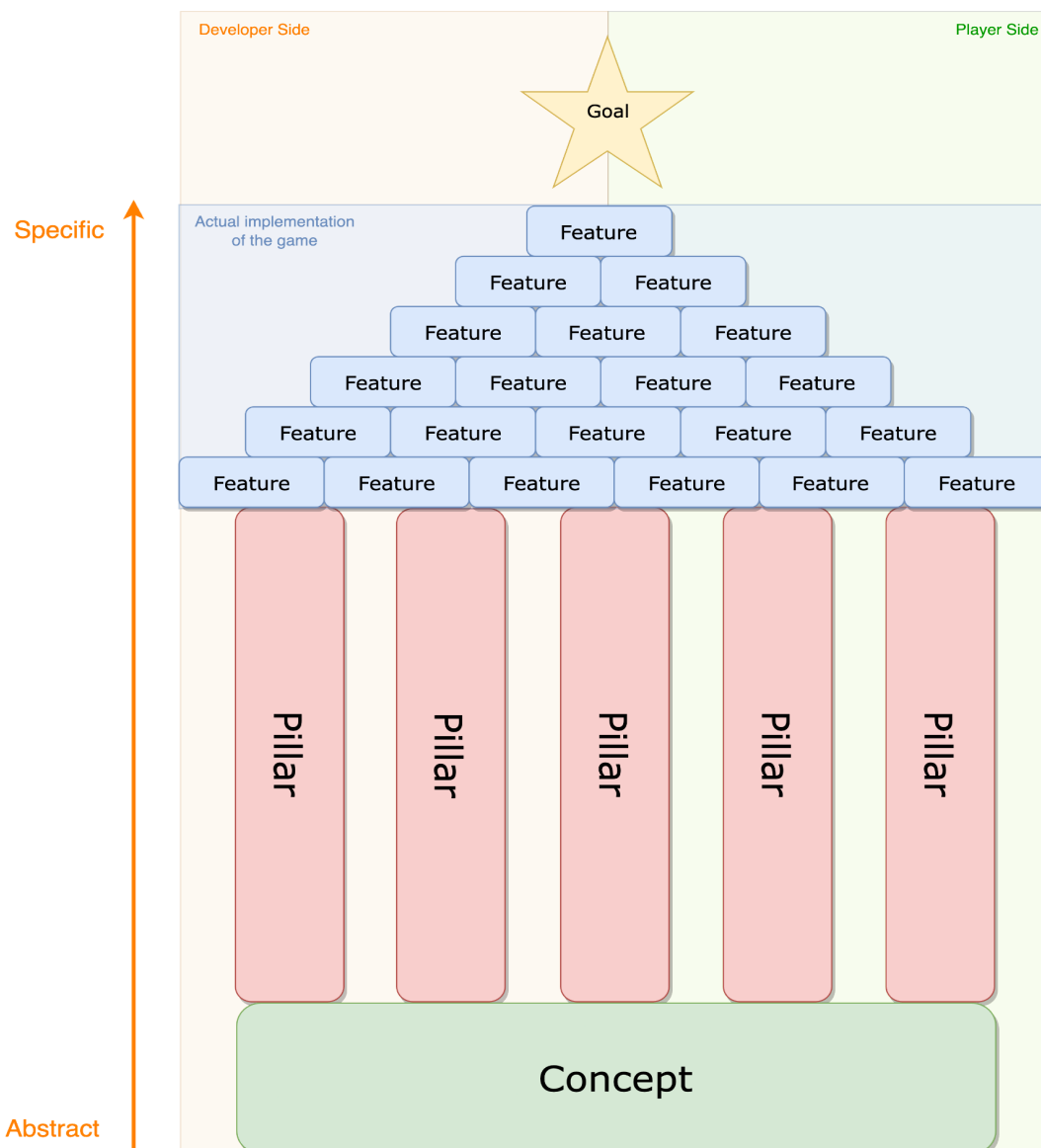


Fig. 1.

Visual representation of the Ludodome's structure

The visual representation of the Ludodome is to be regarded from bottom up. It starts at the highest level of abstraction with the *concept*, only to become more and more specific on the higher levels. The *pillars* emerge from the roots of the *concept*, providing support and stability for the *features*, which represent the *actual game implementation* of the game. At the very top lies the *goal*, which is intentionally not directly connected to the rest of the structure to symbolize that even though all the other components are guided by it, it is never quite reachable. Illustrated in the background are the *developer side* and *players side* to highlight that oftentimes these two entities may see the same product from different angles. In the following section we will take a deeper dive into each of the major components of the Ludodome.

## 4.4. Concept

### 4.4.1. Definition & Goals

“The game begins with an idea.” (J. Schell<sup>[2]</sup>, Chapter 6). Coming up with a good *game concept* is not only one of the first steps a game designer will take, but also one of – if not *the* – most important one, because everything else will be built on top of it. It represents the core idea of a game, the easy-to-understand vision<sup>[13]</sup>.

The *concept* is unique and should be short enough to leave room for creativity, but also complex enough to encapsulate the environment and aesthetic of the game, immersing the reader in the vision of the designer. A phrase should suffice in most cases. Ideally, the concept should include hints about the following game aspects:

- aesthetics (e.g. sensory cues etc.)
- space-time setting (e.g. medieval fantasy/futuristic sci-fi/etc.)
- general atmosphere (e.g. wartime/peacetime/apocalyptic/etc.)

It should be worded in such a way that it resonates with the target audience, because they will be the ones diving into the designer's creation. But it should also resonate with the developers, as they will be the ones bringing the concept to life.

This definition is very similar to how J. Schell defines *themes* in his book “The Art of Game Design: A book of lenses”<sup>[2]</sup>. However, I believe the term *concept* is a more powerful word and thus is more suitable for underlining the importance of its meaning.

The goal of the *game concept* is to tie all the elements of the game together and act as the focal point of all design decisions. In Schell's words, the design team should use every means possible to reinforce the theme – or, in our case, the concept. The concept will also be the source from which the overarching pillars will be derived, which we will discuss in section 4.5. *Pillars*.

#### 4.4.2. Examples

##### Example 1

Schell also provides us with an example in his chapter on themes:

*“The fantasy of being a pirate.”<sup>[2]</sup>*

This is a perfect example of the power of simplicity. Even though this game concept is very short, it reveals a lot of valuable information:

- The term “pirate” alone brings a lot of sensory cues with it: exploration of exotic islands (visual), the sound of waves hitting the ship (acoustic), the smell of sea salt and sweat (olfactory), the taste of rum (gustatory) and the wind blowing in your face while trying to balance on the the swaying ship (tactile).
- In this case, the word “fantasy” doesn't directly imply the genre, but rather it refers to the imagination of being a pirate. The question whether the

game will take place in a fantasy setting (imaginary world, huge krakens, mermaids, etc.) or a realistic setting (real-world locations, real-world characters) still remains open.

- The phrase also hints at the temporal setting of the game. The Golden Age of Piracy<sup>[14]</sup> was between 1650 and 1730, so that is most likely the period when the story in the game will take place. Of course, the numbers do not have to be the exact same, but it is important to pinpoint the technological advancements of the various civilizations that existed at that point.
- The general atmosphere will be rather dynamic; while pirates may not be at war in the geopolitical sense, they survive by raiding other ships and coastal villages, following their own principles and agendas.

### **Example 2**

The following example has been created based on a pillar example in A. Andersen's guide on how to write audio design documents<sup>[6]</sup>. We will analyze the pillar itself in the following sections, but for now, here is a potential *concept* from which it could have emerged:

*“The survival of a small community in a post-apocalyptic world.”*

This concept is a little longer than the previous one, expanding the number of pillars we can directly extract from it.

- The visuals would highly depend on the type of apocalypse that struck the world. Classic examples would be a pandemic (The Last of Us<sup>[c]</sup>) or nuclear catastrophe (Fallout franchise<sup>[d]</sup>). A more original vision would be a mysterious event that suddenly surrounded a small village with a dense forest all around overnight, blocking every road leading out of the village. The outside world as they know it seems to no longer exist and cannot be

contacted by telephone or internet. The inhabitants struggle to find a way out and solve the mystery.

- The space-time setting could be virtually any town on Earth during modern times. Keeping the environment realistic to some degree can help create a deeply relatable connection with the player.
- In this case, the general atmosphere is directly addressed in the concept phrase itself: apocalyptic.

## 4.5. Pillars

### 4.5.1. Definition & Goals

Design pillars are a set of core elements that define a game and give it structure. They support the overall concept of the game, making sure it delivers the intended player experience<sup>[7]</sup>. They also help share the designer's vision with the rest of the team.

The pillars are not only the powerhouse of this whole conceptual framework, they are also the backbone of the game. They are an extension of the *concept* and the foundation upon which the features will be developed. For this reason, it is recommended to iterate over the design pillars and get the whole team on board with them before entering the production phase<sup>[5]</sup>. Moreover, the pillars should also be consulted during production whenever there is a decision anyone on the team needs to make<sup>[15]</sup>.

The goal of the pillars is to bind all the features of the game together into a cohesive and consistent unit and ensure that all elements of the game live in harmony with each other. For this to work, there is a very important rule that should never be broken:

***No decision during pre-production, production or post-production should be against a pillar. All decisions should be in line with the vision of the pillars.***<sup>[5]</sup>

Breaking this rule could potentially lead to breaking the integrity of the game or losing the trust of the players (which is a lot worse).

Design pillars can come in many shapes and forms, but the Ludodome framework differentiates between 3 different major types of pillars, in terms of form: textual, visual and acoustic.

It is important to note that the pillar type strictly refers to the form in which the pillar is created. The type itself does not restrict the elements of the game that the pillar will impact. For example, an acoustic pillar should not only influence the game's music and soundtrack, but it can also be used to provide inspiration for a landscape. This is an issue of syntax-semantics interface, which we will further analyze in section 4.5.5 *Custom Pillars*.

#### **4.5.2. Textual Pillars**

Text is the most straightforward way of creating a pillar and encodes the least amount of information. In its simplest form, it requires no additional tools other than imagination, a pen and paper. An important thing to note about textual pillars is that they should be written in prose and technical terms should be entirely avoided. The technical part will be represented in the next section, 4.3 *Features*, but for now, the goal is to create an emotional connection with the reader<sup>[6]</sup>.

In terms of size, textual pillars can be as short as a phrase, but should not get longer than a paragraph. The more elements are added to a pillar, the more conditions it

will require to be satisfied by a feature, which is not ideal – the more requirements a pillar has, the harder it will be to find a feature to reinforce it.

A. Andersen provides an exemplary textual pillar in his article on game documents<sup>[6]</sup>:

***“The Sound of a Worn-Down World that Evolves With You***

*Our small community is among the few that still remain on Earth. Our world is overrun with fantastical creatures and supernatural events that can change the surroundings that we inhabit, but, in the end, we are still fighting for our very existence with the weapons and tools that we scrounge and forge together.”*

The reason why this pillar is so good is because it can actually be split into multiple different pillars that cover a wide array of aspects of the game:

- The game is about a small community’s survival
- The story takes place on a “worn-down” Earth
- The world is populated with fantastical creatures
- Supernatural events and creatures can shape the surroundings
- Weapons and tools can be forged

Another great set of textual pillars was introduced at GDC<sup>3</sup> 2016 by Marcin Przybyłowicz and Mikolai Stroinski in their presentation on the Slavic Adaptation of Music in ‘Witcher 3’<sup>[el[16]</sup>:

- The Witcher universe is heavily grounded in slavic folklore
- Dark fantasy setting
- Game’s main feature - gritty, mature, engaging
- Both intimate and epic in storytelling

---

<sup>3</sup> Game Developers Conference



- Nothing is good or bad, everything is in the gray zone
- Choices and consequences
- Nonlinearity

Even though this pillar was designed to guide the audio production of the game, it provides very strong pillars for the whole ambiance of the game. It also underlines how interconnected these pillars are and how deep their roots reach throughout the entire game. Nonlinearity, for example, is a term that is normally only used when designing the storyline, but in the case of ‘Witcher 3’, the sound designers went so far as to even incorporate this aspect in the soundtrack as well.

### **4.5.3. Visual Pillars**

Visual pillars are very expressive elements designed to evoke certain emotions by visual means. They are used when words are not enough to express the aesthetic vision of intangible qualities of a design<sup>[17]</sup>. Visual pillars are a great reference for the whole team throughout the whole development process – they help everyone visualize the end result. As a byproduct, they are also a very effective way to explore the visual style of the game design project<sup>[18]</sup>. The process of creating visual pillars alone can trigger inspiration for new ideas. They can either take the form of a simple image or a set of images, commonly called a mood board.

A mood board is a type of collage consisting of diverse visual materials, such as images or samples of objects in a composition. They are used to convey a general idea or feeling about a particular topic<sup>[19]</sup> and are a great support for early design thinking<sup>[20]</sup>.

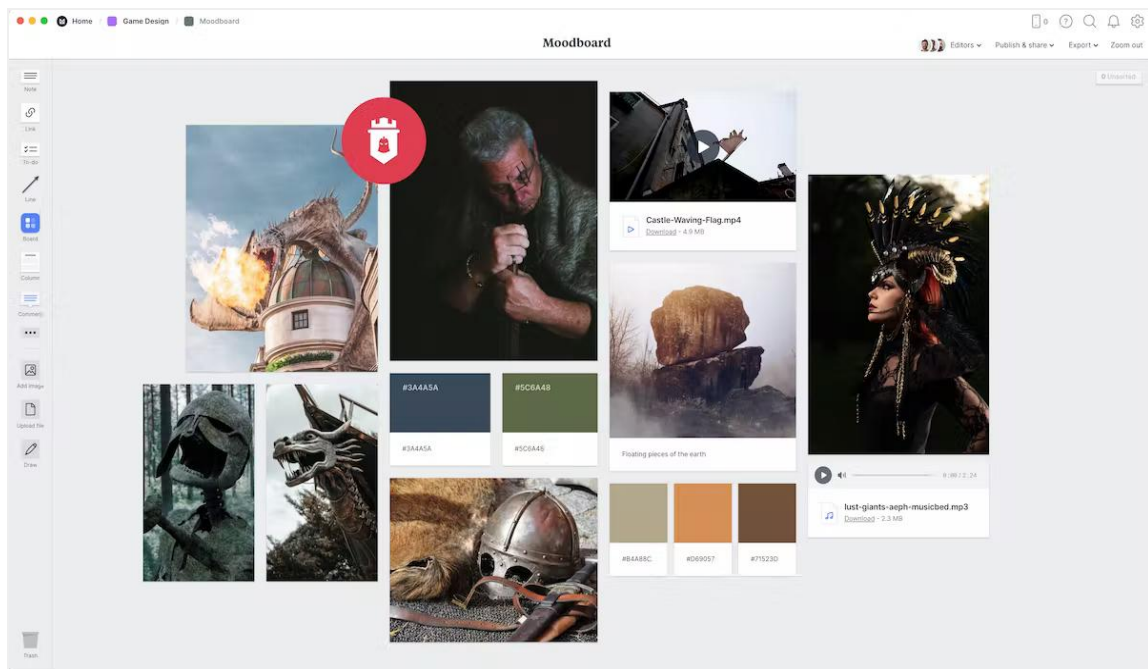


Fig. 2.  
Mood board example by Milanote<sup>[f]</sup>.

Creating visual pillars obviously requires more effort than textual pillars, and it may seem intimidating to some people, especially if they are not necessarily artistically inclined. However, there are plenty of modern tools out there to facilitate this process. Miro<sup>[g]</sup> or Milanote<sup>[f]</sup> are two examples of tools designed to do just that – help designers organize ideas and projects into visual boards. Virtual mood boards also support acoustic pillars – which we will tackle in the next section of this chapter – to add yet another layer of complexity.

Creating custom, unique images of your design vision can prove to be a bit more difficult. Normally, designers would commission artists to carry out this work for them. This generally takes a lot of time, ranging from a few days to a few weeks, or even months. And there is no guarantee that the artist will manage to capture the designer’s vision on the first try – oftentimes the artwork will need to be recommissioned. But

nowadays this process can be significantly simpler. Since 2022, an array of AI text-to-image generators have been made available to the public, like DALL-E 2<sup>[h]</sup> or Midjourney<sup>[i]</sup>. Most of these tools are incredibly easy to use and allow people to create AI-generated images based on a prompt<sup>4</sup> input by the user. They save the designer an immense amount of time and allow him to further explore their own design visions.

Next, we will analyze some images created by Midjourney, which ranks as one of the most popular AI image generators to date. Midjourney is operated through a Discord server and uses Discord bot commands to generate images. The following images have been generated with Midjourney:

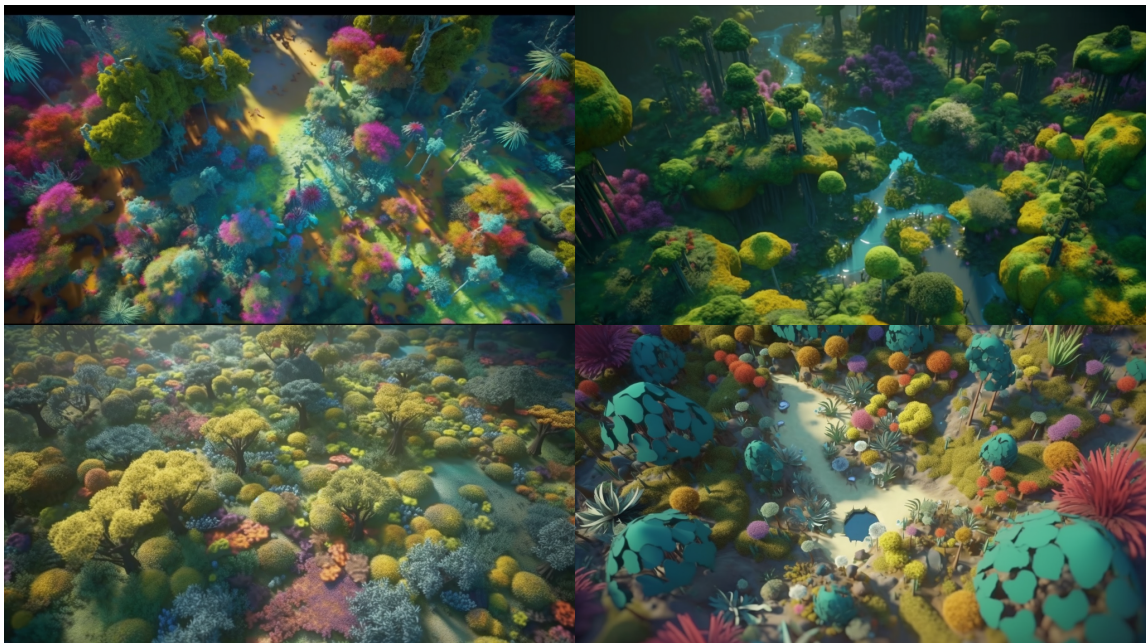


Fig. 3.

AI-generated image of an alien park in aerial view by Midjourney<sup>[i]</sup>.

---

<sup>4</sup> user-generated string containing a command, followed by a description of what is to be generated, followed by a set of parameters

The above image was generated with the following prompt:

```
"/imagine Aerial view, 2d high quality art style. Concept art of an alien natural park. exotic plants, unique animals, photorealism, hyper detailed, vivid colors, realistic high detail 4k render, vibrant volumetric lighting, strong contrast, ray tracing --aspect 16:9 --v 5"
```

The keyword “/imagine” is the bot command created by the Midjourney developers. It is then followed by a somewhat detailed description of the image to be generated. Image details, art style, lighting technique and resolution can all be concatenated by commas, as shown. At the end, we can input some optional parameters to further tweak the final result. In this case, the aspect ratio was specified, along with the model version.<sup>5</sup>

As you can see, the resulting image is not a perfect representation of our prompt. For example, we specified “unique animals”, but there are no animals in the generated image. This approach also requires some additional iterations, but since the generation of an image takes less than 60 seconds, this should not be a problem.

We have now seen the aerial view of our world, but let’s try to see how a first-person view would look like. We want to use 3D art style this time, and because we will now be on the ground, we want to see a beautiful sky with 3 suns. Let’s adapt our prompt accordingly:

---

<sup>5</sup> Further commands and parameters can be found in Midjourney’s official documentation<sup>[21]</sup>.

"/imagine first person view, 3d high quality art style.  
Concept art of an alien natural park. exotic plants, unique  
animals, pastel sky with 3 suns, photorealism, hyper detailed,  
vivid colors, realistic high detail 4k render, vibrant  
volumetric lighting, strong contrast, ray tracing --aspect 16:9  
--v 5"



Fig. 4.

AI-generated image of an alien park in first-person view by Midjourney<sup>[1]</sup>.

The colors and general vibe match our first image, but we are still missing the animals and the 3 suns. Let us make another iteration, this time being more concrete about what animal we would like to see – let us choose an 8-legged tiger and see what

we get. For the purposes of testing just how much this image generator can do, we will also change the style of our images to resemble Deviantart<sup>6</sup> style:

```
/imagine 3d high quality deviantart style. Concept art of  
an alien natural park. exotic plants, alien 8-legged tiger,  
pastel sky with 3 suns of different colors, photorealism, hyper  
detailed, vivid colors, realistic high detail 4k render, vibrant  
volumetric lighting, strong contrast, ray tracing --aspect 16:9
```

```
--v 5
```



Fig. 5.

AI-generated image of an alien park with a tiger in first-person view by  
Midjourney<sup>[1]</sup>.

---

<sup>6</sup> DeviantArt is an American online art community that features artwork, videography and photography, launched on August 7, 2000 by Angelo Sotira, Scott Jarkoff, and Matthew Stephens among others

The model has now successfully rendered an alien tiger, as we requested, in 2 of the 4 images it generated, however it does not have 8 legs. Moreover, there is no evidence in any of the images that there would be 3 suns in the sky. By repeating this process a couple of times, tweaking the parameters each time, the designer will eventually get the model to render his true vision of the environment, and then use the result as a visual pillar.

We have now seen how powerful visual pillars can be, but, as is the case with any powerful object, they should be handled with care. Visual pillars are excellent when trying to express a general mood, but when a pillar aims for concrete information, then a textual pillar would be more suitable.

For instance, let's assume we want to design a game about what it is like to be the Pope, so we set the location of our game to Rome. We might now be inclined to find a beautiful picture of Rome to use as a visual pillar. With the Colosseum being one of the most iconic buildings in Rome, it will most probably be included in our visual pillar. The problem now is that this could also erroneously imply other unwanted collateral information, like gladiator fights, which have nothing to do with our game. So, as a rule of thumb when working with visual pillars, designers should always make sure that everything depicted in the image pillar reflects the game vision.

#### **4.5.4. Acoustic Pillars**

Acoustic pillars are used to induce a certain emotion by auditory means. Soundtracks and ambient sounds expand the arsenal of ways in which the player can be stimulated, adding more depth to the overall experience. Acoustic pillars can range from short sound effects to full movie soundtracks – anything that helps encapsulate the vision of the game in such a way that it can be delivered to the creative team.

Crafting acoustic pillars can prove to be somewhat more difficult compared to the other types of pillars, unless the designer happens to be musically savvy. However, there is such a broad range of music available today that almost every emotion has most probably already been musically tackled by now. It is just a matter of finding it. In his book, D. M. Jackson suggests that the vast archive of film scores that have been created over the past 75 years provides a great source of material<sup>[22]</sup>. There are also numerous royalty free music libraries out there that offer countless audio material, like Soundstripe, Pixabay or Storyblocks, just to name a few.

However, D. M. Jackson also warns about a certain risk associated with this approach. Each of these individual music components were written for a context or purpose that may be different to that of the game vision. And so the decision makers of a project may become fixated on a single piece of music that blends perfectly with their vision and thus becomes the ideal piece in their minds. Once this happens, it is hard for the piece to be replaced, no matter how good the new composition may be. As a solution to this, the author proposes aggregating more musical pieces together to reflect a unique game vision (Jackson, D. M, Chapter 19)<sup>[22]</sup>.

Unfortunately, there is no “AI-shortcut” for creating audio pillars available to the public *just yet*. However, there is a very recent paper by Google Research that tackles this topic precisely. A. Agostinelli et al. introduce a new model called MusicLM<sup>[23]</sup>, which can generate high-fidelity music from text descriptions such as “*a calming violin melody backed by a distorted guitar riff*”.

Moreover, MusicLM is able to combine multiple prompts for fixed timeframes and morph them together into a single audio piece, for example:

*jazz song (0:00-0:15)*

*pop song (0:15-0:30)*

*rock song(0:30-0:45)*



*death metal song (0:45-1:00)*

*rap song (1:00-1:15)*

*string quartet with violins (1:15-1:30)*

*epic movie soundtrack with drums (1:30-1:45)*

*scottish folk song with traditional instruments (1:45-2:00)*

The resulting sample, along with a couple of other examples, can be accessed on the project's webpage<sup>[24]</sup>. Even though the model is not yet publicly available, it has certainly paved the way towards an easily accessible AI-based text-to-music generator. Such a tool would significantly facilitate the creation of audio pillars for game design.

#### **4.5.5. Custom Pillars**

In linguistics, the syntax-semantics interface<sup>[25]</sup> studies the interaction between syntax and semantics and its goal is to explain correlations between form and meaning. A similar theory can be applied to pillars, too, when it comes to the interrelation between form and meaning. We have already established individual pillar types, but combining them together in different ways can drastically alter their semantic meaning, creating new, custom pillars as a result.

While textual, visual and acoustic pillars can be applied to almost any kind of game project, custom pillars are more situational in nature. The aforementioned pillar types can be concatenated to create custom pillars to accommodate various game features and/or mechanics. RPGs<sup>7</sup>, for example, put a great emphasis on a few main characters in the game. The player experiences the game through the lens of these characters. Therefore, it is recommended that these characters all have their own pillars in order to make them consistent, distinctive and complex. The same can be said about games with multiple factions. Each faction should be unique in the game's universe in

---

<sup>7</sup> Role-playing games

order to justify their existence. To enable this, each faction should have its own pillar/s. Common game elements that – if part of the game design – should typically have a dedicated custom pillar include, but are not limited to:

- *Characters* - predominantly textual and visual pillars; acoustic pillars are optional, but very beneficial for the main character/s (e.g. voice tone);
- *Factions* - predominantly textual and visual pillars; acoustic pillars are optional, but a “nice touch”;
- *Locations* - predominantly textual and visual pillars; acoustic pillars are optional;
- *Soundtrack* - predominantly acoustic pillars, optionally reinforced by textual pillars, but visual pillars can also impact audio elements;
- *Gameplay & Tutorial* - predominantly textual pillars, reinforced by visual pillars (e.g. simple prototypes);

This is just a brief list of game components that usually need their own pillars, but, as previously mentioned, this list greatly varies. Depending on the type of game that is being considered the necessary pillars might look entirely different. Generally speaking, every game element that is considered to play a central role in the game in question could – and probably *should* – have its own dedicated pillar.

#### **4.5.6. Unique Pillars**

In principle, anything can be a pillar, as long as it helps expand on the initial game concept and it supports the decision making process on features and gameplay elements. Unique pillars encompass all other pillars that break the conventional classification of textual, visual and acoustic.

Some games are privileged with having great unique pillars right from the start. This is the case with games that are based on books or movies. In situations like these,

an immense source of material for the game is already provided, making it a very robust, unique pillar. A good example of this is ‘The Witcher’<sup>[e]</sup> game, which is based on a book series bearing the same name, written by Andrzej Sapkowski<sup>[26]</sup>. Even though the storyline of the game series is designed to be a spinoff of the events happening in the books, CD Projekt Red have successfully recreated the dark fantasy setting portrayed in A. Sapkowski’s fictional world. Particularly notable is the impeccable conservation of the book series’ eastern european culture in the game. The landscapes, flags, coats of arms and even place names all sound and look very familiar to the Slavic nations<sup>[27]</sup>. Moreover, the characters show predominantly eastern european traits and personalities, peppered with the typical dry humor. Of course, this is partly a result of the game studio being Polish, but achieving this level of detail can only be accomplished by paying close attention to the already provided literature, carving the pillars out of it and staying true to them throughout the entire production process.

Below is another remarkable example of an unconventional, yet ingenious technique that BlueTwelve Studio has employed for their game, Stray:

‘Stray’<sup>[j]</sup> is a third-person adventure game where the player controls a cat, jumping across platforms, climbing obstacles and interacting with other characters. The developers were so dedicated to reproducing the genuine behavior of a cat in their game that they actually brought real cats into the office as a source of inspiration – making them a design pillar. In her article based on an interview with the developers, G. Levine reveals how the office cats helped to create the game.

*“The cats were an invaluable point of reference for the developers, the team studied their behavior to better incorporate it into the game. For example, the game’s lead animator took reference videos of the second cat, a sphinx named Oscar, to help nail the animal’s movements in-game.”*<sup>[28]</sup>

In fact, populating the office with the established pillars is an incredibly effective technique to make sure the team always has them on their radar. A. Andersen endorses

hanging printouts of pillars in the workplace as a constant reminder of where the work should be headed<sup>[6]</sup>. By doing so, the pillars become deeply rooted in one's consciousness and therefore following them becomes almost instinctive.

### **A closing note**

As a closing, yet important note, it is worth bearing in mind that pillars, in general, should not necessarily be reproduced in the game with 100% fidelity. They mainly serve as an inspiration source, to materialize the designer's ideas and vision in a way that can be efficiently shared with the creative team.

## **4.6. Features**

### **4.6.1. Definition & Goals**

As part of this approach, a feature is defined as any type of game element that interacts with the players directly. Out of all the building blocks in the Ludodome, the features construct the facade through which the players ultimately experience the game. In other words, the features represent the toolkit that the developer provides for the players to explore the pillars.

Features can range from general game mechanics to types of enemies and multiplayer modes<sup>[8]</sup>. They represent the actual implementation of the pillars, and should always be aligned with them. This also means that features can then be directly used to create user stories<sup>8</sup> during the production phase, further facilitating the development of the game. Ideally, features should also take into account the technical, organizational and monetary restrictions of implementing them, meaning their feasibility should be considered at all times.

---

<sup>8</sup> In software development and product management, a user story is an informal, natural language description of features of a software system.<sup>[29]</sup>

Further on, we will take a close look at how we can derive features from pillars in such a way that they always stay interconnected – and what can happen when they do not.

### 4.6.2. Example

#### Example 1

M. Eff directly provides us with a feature set in his article on “Pillars, Goals, Features & Elements”<sup>[7]</sup>, which coincidentally fits perfectly with the *concept* example about pirates from Chapter 4.4 *Concept*.

- Steer and modify your own privateer vessel;
- Acquire more vessels and strengthen your pirate fleet;
- Set sail towards uncharted lands and claim what you find;
- Get some more misfits to join your gang;
- Engage in open-sea combat with other pirates;

The list in his article goes on and on, however the important aspect to be noted here is how the features describe a very specific element of the game, compared to the pillars that are purposefully left open-ended.

#### Example 2

For the second example, we will once again expand on the example provided by A. Andersen<sup>[6]</sup> to explore how to forge features out of pillars. To showcase this, we will collect the previously extracted pillars and shape them into various potential features:

**Pillar:** The game is about a small community’s survival

Potential features:

- Coop multiplayer component
- Character classes themed by the necessities of a small community in a post-apocalyptic world
  - Hunter, doctor, scout, etc.
- Survival mechanics
  - Hunger management, fatigue management, etc.
- Town building mechanics
  - By progressing through the game, players can upgrade their town (community)

**Pillar:** The story takes place on a “worn-down” Earth

Potential features:

- Story-driven campaign
- Role-playing game component
  - Character customization & progression
  - Ability tree
- Various (side) quests where players rehabilitate defunct facilities (power plants, dams, etc.) to allow the community to evolve

**Pillar:** The world is populated with fantastical creatures

Potential features:

- Wide range of creatures with different powers (bestiary) that players can interact with
  - Evil beasts have to be fought and exiled from Earth
  - Good beasts can be tamed and used as a wayfinding<sup>9</sup> mechanism

---

<sup>9</sup> Wayfinding in video games refers to the process by which players navigate and orient themselves within the game environment.<sup>[30]</sup>

**Pillar:** Supernatural events and creatures can shape the surroundings

Potential features:

- Wide range of events with various consequences to the community
  - Events are random to avoid predictability
- Changing the surroundings will greatly affect the gameplay dynamic
  - E.g. lakes can change into mountains, forcing players to turn their fisheries into mines

**Pillar:** Weapons and tools can be forged

Potential features:

- Wide range of weapons with various abilities
- Resource-gathering mechanics
  - Resource value increases as players explore the map
- Crafting mechanics

This is just a brief example of how a few well-structured pillars can reveal multiple features. In theory, features can be *almost* infinitely refined, becoming more and more fine-grained with each iteration. However, it eventually comes to a certain point where it no longer makes sense to further cut a feature into smaller parts, similar to how the atom is the smallest recognized division of a chemical element. Once a feature reaches this point, it means it can be officially transformed into a user story, as user stories are coincidentally also considered to be the smallest unit of work in an agile environment<sup>[31]</sup>. This is the point where the Ludodome approach and the agile framework touch base to reveal a highly efficient synergy.

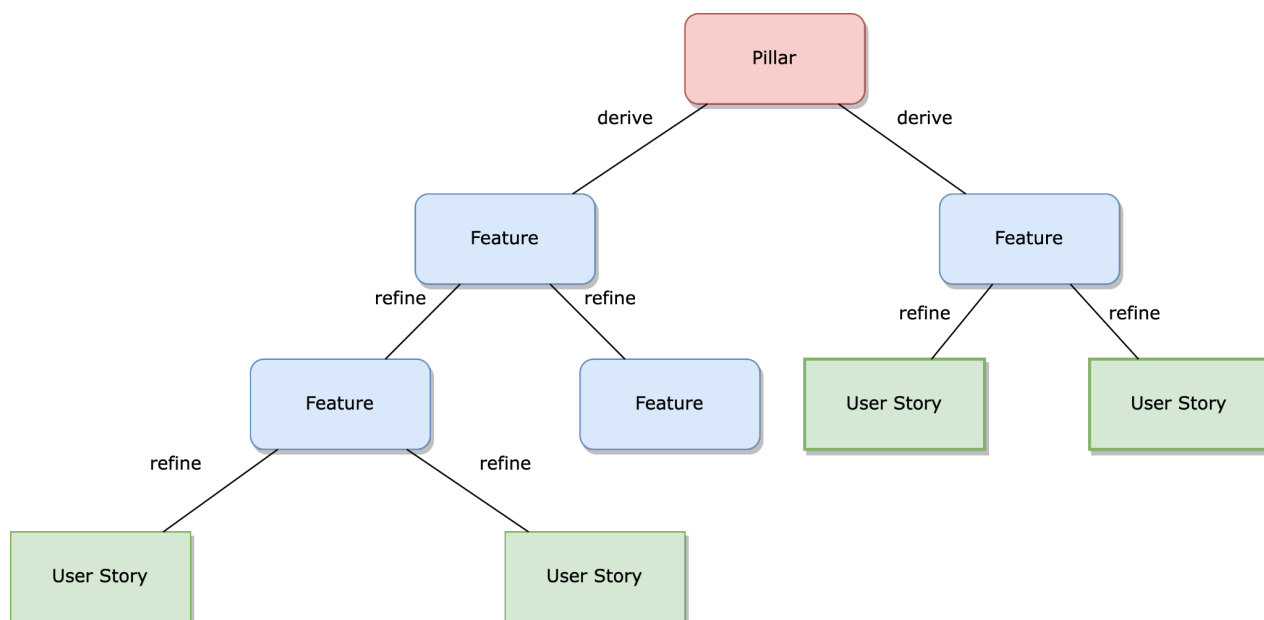


Fig. 6.

Tree diagram that shows how features can be split multiple times until they reach their smallest division and thus become user stories.

### 4.6.3. Features vs. Pillars

Pillars and features may seem quite similar at times. For this reason, it is important to highlight some key differences at this point. One way to differentiate between the two is by asking “how”/”why” questions about the game. In general, the answer to a “why”-question will be a pillar, while the answer to a “how”-question will be a feature. For example:

**Question:** *Why* are players fighting fantastical creatures?

**Answer:** Because their community is fighting for survival in a worn-down Earth.  
(pillar)



**Question:** *How* are players fighting fantastical creatures?

**Answer:** By gathering resources, forging weapons and taming other creatures to help them. (features)

But even by using this technique, some cases may still remain somewhat ambiguous. Therefore, another disambiguation approach is to think of pillars as something that could potentially become part of the game in one form or another. Features, on the other hand, are *certain* to become part of the game in the exact form they are formulated. Generally speaking, pillars provide inspiration, while features provide actual gameplay.

### **What happens when features contradict pillars**

As previously mentioned, one of the most important rules of the Ludodome approach is that the features should always be in line with the pillars. This does not mean that every single feature should be perfectly aligned with every single pillar – in fact, this is quite impossible to achieve in most cases. More importantly, it is crucial that no feature is created that works directly *against* a pillar. Doing so endangers the integrity of the game, as well as the reputation of the developers and can cause serious consequences, as we will see in the following example.

Heroes of Might and Magic III<sup>[k]</sup> is a PC fantasy turn-based strategy PC game developed by the 3DO Company in 1999. It is the third installment of the series with the same name. In the core game, players can choose to play as one of eight different factions (towns) and gain control of a number of heroes that command an army of creatures heavily inspired by medieval fantasy. The game is much more complex, but these are all the key elements for the purpose of this example.

The game saw great success following its release, ranking as the second-best-selling computer game of March 1999 in the United States<sup>[32]</sup> and scoring

remarkable reviews<sup>[33]</sup>. Following this success, the developers began design work on an expansion pack later that year, called *Armageddon's Blade*. As part of this expansion, a new faction – Forge – was planned for release. It was a predominantly high-tech, futuristic town, designed in a steampunk-science-fiction style.

Soon after the announcement, early artwork and sketches of the new town were showcased and the company was immediately hit with a strong community backlash, which involved threats of boycotting the game and even death threats towards the developers. Fans all over the globe were simply enraged about the introduction of sci-fi elements in the game's fantasy setting and were demanding the immediate removal of the Forge. Following the community's uproar, the developers eventually gave in and scrapped the Forge from their plans, replacing it with another faction that was planned for a second expansion pack.<sup>[34][35][36]</sup>



Fig. 7.

Side-by-side comparison between the concept art of a unit from the core game (archer<sup>[37]</sup>, left) and a unit from the unreleased Forge faction (cyber zombie<sup>[35]</sup>, right).

The comparison above makes the thematic differences between the two styles quite apparent. Combining bows and swords together with chainsaws and flamethrowers was deemed too out of place and even distasteful by many fans.

The reason why this specific example was used is because it truly proves how important game integrity is to players. The fans did not wait for the expansion to be released, or even for some gameplay demo to be published. As soon as they saw the concept art, the fanbase felt that adding high-tech sci-fi elements (*features*) to a medieval fantasy game would fundamentally violate the core principles (*pillars*) of the initial *concept*. And this is precisely why the Ludodome framework strongly suggests that features should never transgress any of the pillars.

## **4.7. Goals**

### **4.7.1. Definition & Goals**

Compared to pillars and features, design goals tend to be more abstract. They represent the big picture vision of the game. They are usually not reachable in the absolute sense, yet all design decisions should strive towards achieving that goal<sup>[5]</sup>. In simple terms, the goal of the game should embody the main factor that motivates people to keep playing it<sup>[38]</sup>.

The goal serves as the “north star” for developers and players alike, in the sense that it is always a point of guidance, but never really reachable.

Design goals should be short and clear, and a game should not have more than two or three goals, in order to avoid diluting their importance and help focalize the development process in a unitary direction.

### 4.7.2. Examples

The Ludodome approach classifies design goals in three different categories:

- Continuous goals
- Fixed goals
- Serious goals

Continuous goals are usually used by long-lived games – games that are perpetually updated and rebalanced so as to keep the player involved seemingly forever. A good example for this is League of Legends<sup>[1]</sup>, a MOBA<sup>10</sup> game developed by Riot Games in 2009, where ten players are matched into two opposing teams. Once matched, players may choose one of 163 different characters available in the game<sup>11</sup>, called champions, to control during the game. Every year, players attempt to climb a leaderboard divided by ranks. Players may advance by accumulating LP (League Points) for each won game, while losing LP for defeats. The leaderboards are reset every year, and then a new ‘season’ starts. This system, combined with the fact that the developers actively aim at keeping the average win rate of games at 50%<sup>[39]</sup>, is designed to keep the players invested in the game for long periods of time.

---

<sup>10</sup> Multiplayer online battle arena

<sup>11</sup> As of May 2023



Fig. 8.

Screenshot of a player's promotion to what is considered one of the higher ranks in the game – the Diamond Division

But resetting the leaderboards every year is not the only technique Riot Games uses to keep players hooked to the game for as long as possible. A so-called 'Summoner Level' is visible on the left side of the figure above (the player is level 429). This level is uncapped, meaning players can infinitely advance in level, showcasing their progress on their profile pages as a token of experience with the game. The right side of the screenshot shows the champion mastery, ranked 1 through 7 for each of the 163 (and counting) champions available in the game. Furthermore, the developers constantly change, rebalance and add new elements to the game through patches as often as every two weeks<sup>[40]</sup>. Jeremy Lee, the executive producer, has already confirmed they will likely never stop bringing out new champions to the game<sup>[41]</sup>. With all this in mind, the goal of the game becomes clear: Riot Games wants League of Legends to be a continuously expanding game, making it more and more complex with each patch, all while constantly incentivizing people to keep playing it.

Fixed goals, on the other hand, tend to be more popular with standalone, story-driven games. This is to be expected in a way, as well-rounded stories should generally have a clear exposition, turning point, climax and resolution. The latter is particularly important, as it lies closest to the goal, but since goals are not measurable, it remains inconclusive whether the intended goal was reached or not. A popular example of this is the ‘Dark Souls’<sup>[m]</sup> series, or the more broadly used term “Souls Games”, which include ‘Bloodborne’, ‘Sekiro: Shadows Die Twice’ and ‘Elden Ring’, all developed by the Tokyo-based company FromSoftware. It is known throughout the community that these games are all designed to have a particularly high level of difficulty – so much so, that they influenced the creation of their own subgenre: Soulslike<sup>[42]</sup>. In an interview, Dark Souls creator Hidetaka Miyazaki explained why FromSoftware games are so difficult: “Hardship is what gives meaning to the experience. It’s our identity”. Thus, the high difficulty of the games is by no means coincidental, but clearly intended. “I just want as many players as possible to experience the joy that comes from overcoming hardship.”<sup>[43]</sup> This statement clarifies that FromSoftware actively pursues creating hard games as a fixed goal.

Lastly, there are the serious goals. They are defined as having a primary purpose other than pure entertainment and are the driving force of serious games<sup>[44]</sup>. A classic example is the late 2007 ‘Wii Fit’<sup>[n]</sup> video game<sup>12</sup>, which features a variety of yoga, strength training and balance mini-games and comes with a unique platform peripheral that the player stands upon during play<sup>[45]</sup>. The initial goal of the game was, as stated by the developers “to help get families exercising together”<sup>[46]</sup>. However, the scope of the game eventually went even further than this, with the Wii Fit being used for physiotherapy rehabilitation in children<sup>[47]</sup>.

A more popular and accessible game with what can be considered a serious goal is ‘Pokémon Go’<sup>[o]</sup>, an augmented reality mobile game developed by Niantic and released

---

<sup>12</sup> Also called exergame – video games that are also a form of exercise<sup>[48]</sup>

in 2016. In this game, players use GPS technology to locate and catch “pocket monsters” (called Pokémon) in different real-world locations. The game had a unique goal: it was designed to encourage people to explore their surroundings in real life, and it did just that with great success. Not only has Pokémon increased the daily walking distance of people all around the globe<sup>[49]</sup>, but it was also widely acclaimed for bringing people together and it was described as “more than just a game” by Forbes<sup>13</sup>: “It’s a phenomenon that people never saw coming”<sup>[50]</sup>.

This being said, it’s absolutely acceptable for games to have the simple goal of “just having fun” – many successful games do – but having a unique goal can prove to be very beneficial and may give your game a competitive edge over others.

#### **4.8. The Ludodome in practice**

Now that all the building blocks of the Ludodome have been defined, we need to put them all together in a way that can be efficiently used in practice. The following section is based on the *Macro/Micro Design* methodology, first presented by Mark Cerny during the D.I.C.E.<sup>14</sup> Summit in 2002<sup>[51]</sup>. The terminology was also used a few years later during the production of ‘Uncharted 2’<sup>[p]</sup>, developed by Naughty Dog in 2009<sup>[52]</sup>. A reiteration of this approach was presented by Andrew Dotsenko in his article “Game Design Framework: On the way to good design”<sup>[9]</sup>.

---

<sup>13</sup> Forbes is an American business magazine

<sup>14</sup> Acronym for Design Innovate Communicate Entertain

### 4.8.1. Alignment table

The core idea of M. Cerny's framework is rather simple; it starts with "an idea of a great experience"<sup>[9]</sup>, which is very similar to how the Ludodome starts with a *concept*. The design of this idea is then divided into two main types: Marco Design and Micro Design, each on a different abstraction layer.

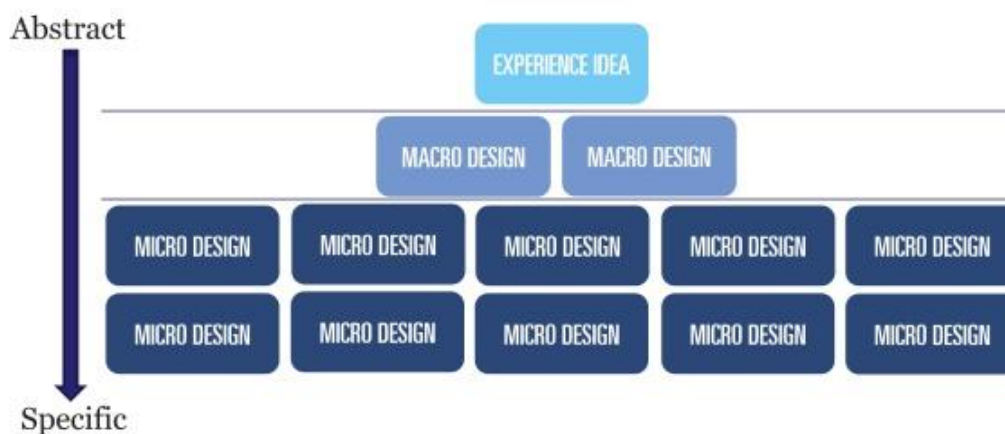


Fig. 9.

Micro/Macro Design overview<sup>[9]</sup>.

The Macro Design represents the concept that describes the player experience on the higher abstraction level. This definition also bears a strong resemblance to the pillars in the Ludodome, which is also why this methodology works so well with our framework.

The Micro Design depicts a more detailed description of the game elements and takes into account all technical restrictions and edge cases. Once again, a striking similarity with our approach emerges: the features.



In order to further explore Macro Design ideas, the developers at Naughty Dog have crafted a so-called Macro Design spreadsheet:

UNCHARTED 2 Macro Design																	
LEVELS	LOOK DESCRIPTION	TIME OF DAY/ MOOD	ALLY-NPC	ENEMY MODELS	MACRO GAMEPLAY	MACRO FLOW	PLAYER MECHANICS										
							Free Climb/Dyno	Wall Jump	Free Ropes	Pendulum	Monkey Bars	Monkey Swing	Balance Beams	Carry Objects Heavy	Carry Objects Light	Traversal Gameplay v.1	
Warzone																	
war-1-market	Nepalese city broken & burning	High Noon - War-torn & smokey		Laz Army HOT Freedom Fighters	Explore Traverse Minor Gunfights	Basic Gunplay Traversal Gunplay	X	X			X	X					X
war-2-streets	Nepalese city broken & burning	High Noon - War-torn & smokey	Chloe-2	Laz Army HOT Freedom Fighters	Explore Traverse Minor Gunfights	Basic Gunplay Traversal Gunplay	X	X			X	X					X
war-3-inside war-4-highrise	Nepalese city broken & burning	High Noon - War-torn & smokey	Chloe-2	Laz Army HOT Freedom Fighters	Explore Traverse Minor Gunfights	Basic Gunplay Traversal Gunplay Get to higher ground (hotel)	X	X			X	X					X
city	Nepalese city broken & burning	High Noon - War-torn & smokey	Chloe-2	Laz Army HOT Freedom Fighters	Explore Traverse Minor Gunfights	Skirt close to Laz Army	X	X			X	X					X

Fig. 10.

Uncharted 2 Macro Design spreadsheet<sup>[53]</sup>.

The Macro Design table is reported to be the key document created as a result of six months of pre-production of ‘Uncharted 2’. The rows in the table usually represent the different levels in the game, ordered in the corresponding progression order from top to bottom. The columns enumerate the level content. This part is highly dependent on the type of game that is being developed, but in general it should contain information about:

- Metadata (duration, scalability, production order, etc.)
- Level Design (level flow, pacing, player goals, mechanics etc.)
- Art (theme, mood, colors, cutscenes etc.)
- Story (plot beats, motivation, characters, locations)
- Soundtrack (ambiance, instruments)

M. Cerny's original methodology<sup>[51]</sup>, as well as A. Dotsenko's<sup>[9]</sup> reinterpretation both delve into further details around this approach, however, for the purposes of this thesis, the information presented so far suffices. In the following, we will reinterpret the Design Macro table to adapt it to the Ludodome framework. More precisely, we will use the overall layout of the table to check the alignment of features with the pillars, as well the overarching concept. This is possible thanks to the parallelism of the two approaches mentioned at the beginning of this section. Thus, using the same general structure of the Macro Design spreadsheet, we will place features in the rows and pillars in the columns. The very last column/s will contain the goal/s, as defined in Chapter 4.7 *Goals*. This result in the following layout:

	Pillar 1	Pillar 2	Pillar 3	Pillar 4	...	Pillar X	Goal
Feature 1					...		
Feature 2					...		
Feature 3					...		
Feature 4					...		
Feature 5					...		
Feature 6					...		
...	...	...	...	...	...	...	...
Feature X					...		

Table 1.  
Empty alignment table.

The alignment table now allows us to “fact-check” our features against our pillars and goals. For every feature-pillar combination, it should be investigated to what extent a certain feature actually implements the corresponding pillar. Analogously, it should also be examined whether the features actively strive towards achieving the goals. The purpose of the alignment table is to visualize the consistency of the game design.

	Pillar 1	Pillar 2	Pillar 3	Pillar 4	...	Pillar X	Goal
Feature 1	✓		✓		...	✓	✓
Feature 2		✓	✗		...		
Feature 3	✓	✓		✓	...	✓	✓
Feature 4	✓	✓			...	✓	✓
Feature 5	✓			✓	...		✓
Feature 6		✓	✓		...	✓	
...		✓			...		...
Feature X	✓		✓		...		✓

Table 2.

Filled alignment table.

Green check mark symbolizes harmony between feature and pillar; red cross means the feature (or some part of it) contradicts the pillar; empty cell means the feature and pillar are neutral to each other.

For the purposes of ease of understanding, the table has been filled with check marks. Normally, the content of the table should be in written form with “enough of a level of abstraction that you don’t waste time devising details that later get changed.”<sup>[52]</sup>

Looking at the filled alignment table can reveal a lot of crucial information about the overall game design:

- The first aspect that comes to attention is the fact that, even though correctly aligned with “Pillar 2”, “Feature 2” is in direct contradiction with “Pillar 3”. This may not mean that the feature should be scrapped entirely, it only highlights that some part of it goes against a pillar, and thus the feature should be reworked.
- Another facet that comes to mind is that “Pillar 4” is rather underrepresented, having only 2 features to reinforce it. Once again, this is not detrimental to the pillar, but it suggests that we may need to add more features in order to make “Pillar 4” shine.
- Other than that, the overall design looks promising: most feature-pillar relations are in harmony and most features actively work towards achieving the goal.

It is important to be noted at this point that the alignment table does not directly imply how good or bad a game is. It purely offers a high-level overview of the design’s consistency.

### **4.8.2. Timeline**

In a presentation at D.I.C.E. 2010, Uncharted 2 co-lead designer Richard Lemarchand highlighted that one of the key factors in the game’s success was the “loosely structured” development process. He described how the game’s pre-production phase lasted a full half-year, devoid of firm deadlines, milestones or any tangible production goals. The focus was rather on getting the whole team together to brainstorm around story ideas, create rough gameplay concepts and “pile up mountains of concept art”<sup>[52]</sup>.

A similar approach should also be applied when working in the Ludodome environment. The concept, pillars, features, goals and alignment table should all be

established during pre-production, before starting the actual development process. Doing so ensures the whole team is given enough creative freedom, encourages cross-disciplinary collaboration and ensures the game design retains a consistent groundwork for the game to be built upon.

Once the Ludodome foundation is laid, it should not be drastically changed once the game goes into the production phase, especially not the pillars. R. Lemarchand remarked during the same presentation that the final game would deviate from the pre-production design by only 5 to 10 percent<sup>[52]</sup>.

## 5. Conclusion

In conclusion, it can be argued that the thesis has achieved its goal of creating a customizable framework to facilitate the documentation and development of modern games. If applied correctly, the *Ludodome* methodology not only provides a strong foundation for the game production, but its resulting artifacts also furnish a consistent, harmonized compendium of documentation, covering almost all types of game documents recommended by J. Schell's in his book "The Art of Game Design: A Book of Lenses" (Chapter 24)<sup>[2]</sup>:

- *The Game Design Overview* and *Detailed Design Document* are both covered by the *alignment table*;
- *The Story Overview* and *Story Bible* are addressed by the *story pillars*;
- *The Pipeline Overview* is taken care of through the *features* extracted from the *visual pillars*;
- *The System Limitations* are not directly covered, but should be considered as part of the process of creating the *features*;
- *The Art Bible* and *Concept Art Overview* are given by the *visual pillars*;
- *The Project Schedule* is also not directly addressed, however the *alignment table* should greatly facilitate the planning of a reliable schedule;
- *Game Tutorial* covered as part of *custom pillars*;
- *Manual* not directly covered, however the framework delivers all the necessary components for creating one;
- *Technical Design Document*, *Game Budget*, *Script* and *Game Walkthrough* remain uncovered;

However, this methodology is yet to be tested before it can be validated as "successful", and plenty of room for improvement still remains. These topics will be further discussed in the next section, 6. *Future Work*.

## 6. Future Work

For this particular thesis, future work is highly recommended. Even though the content itself relies on already individually researched and validated building blocks, the Ludodome still only floats on the theory layer, as it has never before been put in practice as a whole construct. Thus, applying this approach in a practical circumstance is crucial to establishing its worthiness in the industry. A suitable environment for the purpose of testing this framework could be in *teaching*, e.g. as part of a group project, practical course, or even a guided research. It is highly important that feedback is collected throughout the entire testing process, so that the framework can then undergo iterative design<sup>15</sup>.

Further potential future work, perhaps more complex, consists of developing a tool or even a suite of tools to facilitate game development with the Ludodome framework, similar to how agile development heavily relies on tools like Jira<sup>[q]</sup> and Confluence<sup>[a]</sup>. Ideally, such a tool would integrate not only an internal knowledge sharing platform (such as Confluence) and a ticket management system (such as Jira), but also all the other tools presented in this thesis, like moodboards, AI-powered generation aids for art and audio elements, as well as a game design macro table.

---

<sup>15</sup> Iterative design = design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process<sup>[54]</sup>

## Ludography

[a] Atlassian (2004). *Confluence* [Web].

<https://www.atlassian.com/software/confluence>

[b] Blizzard Entertainment (1998). *StarCraft* [PC, macOS, Nintendo 64].

[c] Naughty Dog (2013). *The Last of Us* [PlayStation 3, Playstation 4].

[d] Interplay Entertainment, Black Isle Studios, Micro Forté, Bethesda Game Studios, Obsidian Entertainment (1997-present). *Fallout* [DOS, Microsoft Windows, Mac OS, Mac OS X, Nintendo Switch, PlayStation 2, PlayStation 3, PlayStation 4, PlayStation 5, Xbox, Xbox 360, Xbox One, Xbox Series X/S, iOS, Android].

[e] CD Projekt Red (2015). *The Witcher 3* [PlayStation 4, Windows, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S].

[f] Milanote (2019). *Milanote* [Web]. <https://milanote.com/>

[g] Khusid, A. and Shardin, O. (2004). *Miro* [Web]. <https://miro.com/index/>

[h] OpenAI (2022). *DALL-E 2* [Web]. <https://openai.com/product/dall-e-2>

[i] Midjourney, Inc. (2022). *Midjourney* [Web]. <https://www.midjourney.com/>

[j] BlueTwelve Studio (2022). *Stray* [PlayStation 4, Playstation 5, Windows].

[k] The 3DO Company (1999). *Heroes of Might and Magic III: The Restoration of Erathia* [Windows, Macintosh, Linux].

[l] Riot Games (2013). *League of Legends* [Windows, Mac OS X].

[m] FromSoftware (2011-2018). *Dark Souls (Series)* [PlayStation 3, Xbox 360, Windows, PlayStation 4, Xbox One, Nintendo Switch].

[n] Nintendo (2007). *Wii Fit* [Wii].

[o] Niantic (2016). *Pokémon Go* [iOS, iPadOS, Android].



[p] Naughty Dog (2009). *Uncharted 2: Among Thieves* [PlayStation 3].

[q] Atlassian (2004). *Confluence* [Web]. <https://www.atlassian.com/software/jira>

## List of Figures

Fig. 1. Visual representation of the Ludodome’s structure.....	18
Fig. 2. Mood board example by Milanote <sup>[f]</sup> .....	26
Fig. 3. AI-generated image of an alien park in aerial view by Midjourney <sup>[i]</sup> .....	27
Fig. 4. AI-generated image of an alien park in first-person view by Midjourney <sup>[i]</sup> ....	29
Fig. 5. AI-generated image of an alien park with a tiger in first-person view by Midjourney <sup>[i]</sup> .....	30
Fig. 6. Tree diagram that shows how features can be split multiple times.....	40
Fig. 7. Side-by-side comparison between the concept arts .....	42
Fig. 8. Screenshot of a player’s promotion .....	45
Fig. 9. Micro/Macro Design overview <sup>[9]</sup> .....	48
Fig. 10. Uncharted 2 Macro Design spreadsheet <sup>[53]</sup> .....	49

## List of Tables

Table 1. Empty alignment table.....	50
Table 1. Filled alignment table.....	51

## References

- [1] Baldwin, M. (2005). *Game design document outline*. Baldwin Consultant, 24.
- [2] Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC press.
- [3] Beck, K. and Beedle, M. and van Bennekum, A. and Cockburn, A. and Cunningham, W. and Fowler, M. and Grenning, J. and Highsmith, J. and Hunt, A. and Jeffries, R. and Kern, J. and Marick, B. and Martin, R. C. and Mellor, S. and Schwaber, K. and Sutherland, J. and Thomas, D. (2001). *The agile manifesto*. Retrieved from <https://agilemanifesto.org/>.
- [4] Singh, N. (n.d.). *Understanding the Difference between Template And Framework*. Retrieved from <https://www.techaltair.com/understanding-the-difference-between-template-and-framework/>.
- [5] Dyrda, D. (2020). *Game Design Methods and Tools: Documents in Game Development*. Lecture: Introduction to Audio for Games, Technical University of Munich, Department of Informatics.
- [6] Andersen, A. (2016). *Game Audio Guide: How to Write a (Game-Changing) Audio Design Document Today*. Retrieved from <https://www.asoundeffect.com/game-audio-design-document/>.
- [7] Eff, M. (2023). *Pillars, Goals, Features & Elements*. Retrieved from <https://soundand.design/pillars-goals-features-elements-fe45771b4ff5>.
- [8] Mullich, D. (2018). *Game Mechanics Vs. Game Features*. Retrieved from <https://davidmullich.com/2018/08/06/game-mechanics-vs-game-features/>.
- [9] Dotsenko, A. (2016). *Game Design Framework: On the way to good Game Design*. Retrieved from

- <https://www.gamedeveloper.com/design/game-design-framework-on-the-way-to-good-game-design>.
- [10] Harmon, P., & Wolf, C. (2016). The state of business process management. *BP Trends*.
- [11] Atlassian (n.d.). *The importance of documentation (because it's way more than a formality)*. Retrieved from <https://www.atlassian.com/work-management/knowledge-sharing/documentation/importance-of-documentation>.
- [12] Blizzard Entertainment (1998). *StarCraft*. Retrieved from <http://ftp.blizzard.com/pub/misc/StarCraft.PDF>.
- [13] Pluralsight (2014). *Game Concept: How To Come up With a Game Idea*. Retrieved from <https://www.pluralsight.com/blog/film-games/creating-game-concept-first-step-getting-game-ground>.
- [14] Wikipedia. (n.d.). *Golden Age of Piracy*. Retrieved from [https://en.wikipedia.org/wiki/Golden\\_Age\\_of\\_Piracy](https://en.wikipedia.org/wiki/Golden_Age_of_Piracy).
- [15] Kubik, R. (2020). *Using Design Pillars to Keep Wildfire Swap's Development on Track*. Retrieved from <https://ryankubik.com/blog/wildfire-swap-design-pillars>.
- [16] Przybyłowicz, M. and Stroinski, M. (2016). *Slavic Adaptation of Music*. Retrieved from [https://ubm-twvideo01.s3.amazonaws.com/01/vault/gdc2016/Presentations/Przybylowicz\\_Marcin\\_SlavicAdaptationOfMusic.pdf](https://ubm-twvideo01.s3.amazonaws.com/01/vault/gdc2016/Presentations/Przybylowicz_Marcin_SlavicAdaptationOfMusic.pdf).
- [17] Endrissat, N., Islam, G., & Noppeney, C. (2016). *Visual organizing: Balancing coordination and creative freedom via mood boards*. *Journal of Business Research*, 69(7), 2353-2362.

- [18] Clancy, M. (2023). *How to make a moodboard in 10 easy steps*. Retrieved from <https://milanote.com/guide/game-design-moodboard>.
- [19] Wikipedia (n.d.). *Mood board*. Retrieved from [https://en.wikipedia.org/wiki/Mood\\_board](https://en.wikipedia.org/wiki/Mood_board).
- [20] Rieuf, V., Bouchard, C., & Aoussat, A. (2015). *Immersive moodboards, a comparative study of industrial design inspiration material*. *Journal of Design Research*, 13(1), 78-106.
- [21] Midjourney, Inc. (2016). *Midjourney Documentation*. Retrieved from <https://docs.midjourney.com/>.
- [22] Jackson, D. M. (2003). *Sonic branding*. Palgrave Macmillan Books.
- [23] Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., ... & Frank, C. (2023). *MusicLM: Generating music from text*. arXiv preprint arXiv:2301.11325.
- [24] Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., ... & Frank, C. (2023). *MusicLM: Generating music from text*. Retrieved from <https://google-research.github.io/seanet/musiclm/examples/>.
- [25] Wikipedia (n.d.). *Syntax–semantics interface*. Retrieved from [https://en.wikipedia.org/wiki/Syntax%E2%80%93semantics\\_interface](https://en.wikipedia.org/wiki/Syntax%E2%80%93semantics_interface).
- [26] Sapkowski, A (1990-2013). *The Witcher*. SuperNowa.
- [27] Gawroński, S., & Bajorek, K. (2020, October). *A Real Witcher—Slavic or Universal; from a Book, a Game or a TV Series? In the Circle of Multimedia Adaptations of a Fantasy Series of Novels “The Witcher” by A. Sapkowski*. In *Arts* (Vol. 9, No. 4, p. 102). MDPI.
- [28] Levine, G. (2022). *Stray Developers Talked about The Real Cats That Inspired Them*. Retrieved from

<https://80.lv/articles/stray-developers-talked-about-the-real-cats-that-inspired-them/>.

- [29] Wikipedia (n.d.). *User story*. Retrieved from [https://en.wikipedia.org/wiki/User\\_story](https://en.wikipedia.org/wiki/User_story).
- [30] Eff, M. (2023). *Mapping & Wayfinding*. Retrieved from <https://soundand.design/mapping-wayfinding-5fc21d054898>.
- [31] Rehkopf, M. (n.d.). *User stories with examples and a template*. Retrieved from <https://www.atlassian.com/agile/project-management/user-stories>.
- [32] Fudge, J. (1999). *EA Tops Charts*. PC Data, Reston, Virginia. Retrieved from [https://web.archive.org/web/20050407232621/http://www.cdmag.com/articles/019/039/pc\\_data\\_march99.html](https://web.archive.org/web/20050407232621/http://www.cdmag.com/articles/019/039/pc_data_march99.html).
- [33] Gamespot (1999). *Heroes of Might and Magic III*. Retrieved from <https://web.archive.org/web/20190530064822/https://www.gamerankings.com/pc/63965-heroes-of-might-and-magic-iii/index.html>.
- [34] Fandom (n.d.). *Forge (town)*. Retrieved from [https://mightandmagic.fandom.com/wiki/Forge\\_\(town\)](https://mightandmagic.fandom.com/wiki/Forge_(town)).
- [35] Age of Heroes (n.d.). *Forge Town*. Retrieved from <http://www.heroesofmightandmagic.com/heroes3ab/forgetown.shtml>.
- [36] Heroes 3 wiki (n.d.). *Forge*. Retrieved from <https://heroes.thelazy.net/index.php/Forge>
- [37] The 3DO Company (1999). *Heroes of Might and Magic III Player Manual*. Retrieved from [https://heroes3wog.net/download/\[Heroes%203\]%20Restoration%20of%20Eratia%20Manual.pdf](https://heroes3wog.net/download/[Heroes%203]%20Restoration%20of%20Eratia%20Manual.pdf).

- [38] Gonzalez, L. (2016). *How to Write a Game Design Document*. Retrieved from <https://www.gamedeveloper.com/business/how-to-write-a-game-design-document>
- [39] Majda, P., Moormann, J., Forbes, B. & Kenna, E. (2018). */DEV: MATCHMAKING REAL TALK*. Retrieved from <https://nexus.leagueoflegends.com/en-us/2018/02/dev-matchmaking-real-talk/>
- [40] McIntyre, I. (2023). *Here's every update date on Riot's LoL patch schedule for the 2023 season*. Retrieved from <https://dotesports.com/league-of-legends/news/league-of-legends-2023-patch-schedule/>.
- [41] Díaz de León, A. (2022). *League of Legends' producer confirms Riot will not stop releasing new champions*. Retrieved from <https://www.levelup.com/en/news/709256/League-of-Legends-producer-confirms-Riot-will-not-stop-releasing-new-champions/>.
- [42] Wikipedia (n.d.). *Soulslike*. Retrieved from <https://en.wikipedia.org/wiki/Soulslike>
- [43] Gafarova, O. (2022). *Dark Souls Creator Explained Why FromSoftware Games Are So Difficult*. Retrieved from <https://80.lv/articles/dark-souls-creator-explained-why-fromsoftware-games-are-so-difficult/>.
- [44] Wikipedia (n.d.). *Serious game*. Retrieved from [https://en.wikipedia.org/wiki/Serious\\_game](https://en.wikipedia.org/wiki/Serious_game).
- [45] Wikipedia (n.d.). *Wii Fit*. Retrieved from [https://en.wikipedia.org/wiki/Wii\\_Fit](https://en.wikipedia.org/wiki/Wii_Fit).
- [46] Nintendo (2008). *Part 1 - A Truly Ground-breaking Collection of Games*. Retrieved from



- <https://web.archive.org/web/20080708021300/http://www.nintendo.com/wii/what/iwataasks/volume-4/part-1>.
- [47] Ivor, K. (2009). *Nintendo Wii Fit for physiotherapy*. Retrieved from <https://web.archive.org/web/20091123231417/http://stanford.wellsphere.com/general-medicine-article/nintendo-wii-fit-for-physiotherapy/460917>.
- [48] Wikipedia (n.d.). *Fitness game*. Retrieved from [https://en.wikipedia.org/wiki/Fitness\\_game](https://en.wikipedia.org/wiki/Fitness_game).
- [49] Grebey, J. (2016). *'Pokemon Go' Players Have Walked Enough to Reach Pluto*. Retrieved from <https://www.inverse.com/article/25517-pokemon-go-players-total-distance-walked-2016>.
- [50] Kain, E. (2016). *'Pokémon GO' Is More Than Just A Game And It's Bringing People Together*. Retrieved from <https://www.forbes.com/sites/erikkain/2016/07/11/pokemon-go-is-more-than-just-a-game-its-a-phenomenon-thats-bringing-people-together/#5f4df9864743>.
- [51] Academy of Interactive Arts & Sciences (2012). *D.I.C.E. Summit 2002 - Mark Cerny*. Retrieved from [https://www.youtube.com/watch?v=QOAW9ioWAvE&ab\\_channel=AcademyofInteractiveArts%26Sciences](https://www.youtube.com/watch?v=QOAW9ioWAvE&ab_channel=AcademyofInteractiveArts%26Sciences).
- [52] McInnis, S. (2010). *Naughty Dog designer maps out Uncharted 2 development*. Retrieved from <https://www.gamespot.com/articles/naughty-dog-designer-maps-out-uncharted-2-development/1100-6251473/>.
- [53] Leadbetter, R. (2015). *Among Friends: How Naughty Dog Built Uncharted 2*. Retrieved from

<https://www.eurogamer.net/digitalfoundry-among-friends-how-naughty-dog-built-uncharted-2>.

[54] Wikipedia (n.d.). *Iterative design*. Retrieved from [https://en.wikipedia.org/wiki/Iterative\\_design](https://en.wikipedia.org/wiki/Iterative_design).