Why MDA?

The pursuit of a Game Design Onthology

Rogerio Candido da Silva Junior

Submitted to University of Beira Interior in partial fulfilment of the requirement for the degree of Master in

Design and Development of Digital Games

Supervised by: Doctor Frutuoso Silva

January, 2021

The tittle of this Thesis should be read in the same musical tonality as in the chorus of the song YMCA. And the same dance moves. Except the "C" part of the song - you change for "D". The "D" is really important. No pun intended.

The MDA acronym in this thesis does not stand for Methylenedioxyamphetamine.
Unfortunately.

Acknowledgements

Despite being a difficult year, I was blessed by having support during this process. I cant thank enough, and I cant thank them all. But I would like to acknowledge the support of some here, and send a thanks:
To my mom for being supportive in every decision that i made - even when she probably should have not;
To my father, who presented me with video games every Christmas during my childhood;
To the Triad, who helped me not feeling lonely when I was alone;
To professor Frutuoso, who was really supportive during this difficult year;
To UBI and its teachers, who gave me enough knowledge that wouldn't fit in a hundred thesis;
To my master's colleagues, for being there for me during difficult times and partying times, and of course, during learning times;
To the city of Covilha, where I was so well received, made so many friends and created so many memories;
To Classplash and my colleagues, for being supportive whenever I needed some time off during this Thesis;
To Cipsoft, for creating the game that has sparked the will of developing games in me;
Last but not least, special thanks to my friends Ianka Freitas and Arthur Teles for being present even when being far away.

Resumo

Ontologia é um conjunto de conceitos e categorias que representam um domínio. Houve muitas tentativas de se criar uma ontologia amplamente aceita para o domínio de desenvolvimento de jogos. A maioria delas é definida com base em uma perspectiva analítica: poucas têm encontrado uso frequente fora das universidades, pois não são facilmente traduzidas para o desenvolvimento de jogos – numa perspectiva de design. Existem alguns aspectos centrais do domínio que dificultam a realização dessa tarefa. Além disso, os designers de jogos tendem a recusar uma metodologia ou uma forma estruturada de desenvolver um jogo – a principal preocupação é que isso pode prejudicar a criatividade num campo que não poderia sobreviver sem ela. Uma ontologia definida melhoraria e amadureceria a crescente indústria de jogos digitais, tanto melhorando a compreensão do domínio quanto apoiando uma metodologia estruturada para desenvolver jogos. Novas ontologias melhoram a resolução de problemas no domínio.

Esta tese irá descrever aspectos de jogos digitais e mostrar como eles dificultam a criação de uma ontologia para o domínio, principalmente quando se trata de uma perspectiva de design. Seguirá analisando o mais próximo de uma ontologia unificada que existe para o domínio do jogo: a estrutura de Mecânicas, Dinâmicas e Estéticas, ou MDA. Ao revisar o estado da arte em torno desta *framework*, identificou-se lacunas dentro do MDA que prejudicam seu uso em uma perspectiva de design, como a falta de suporte para alguns dos aspectos descritos dos jogos digitais e como a indústria afirma que uma metodologia estruturada não suporta a criatividade – uma característica central do processo de design do jogo.

Para reduzir essas lacunas, são propostas alterações nas taxonomias do MDA – proporcionando um melhor uso da mesma dentro da perspectiva de um designer, objetivo alcançado ao absorver os aspectos de design do domínio e ao superar as dificuldades encontradas na literatura atual do domínio de jogos.

Palavras Chave

MDA, Desenvolvimento de Jogos, Ontologia de Jogos, Metodologia de Desenvolvimento de Jogos, Mecânicas de Jogos, Dinâmica de Jogos

Abstract

An ontology is a set of concepts and categories that represent a domain. There have been many attempts into creating a widely accepted ontology for the game domain. Most of them are defined based on an analytical perspective: few have found frequent use outside universities, as they are not easily translated to the development of games – a design perspective. There are some core aspects of the domain that harshen the achievement of this task. Not only this, game designers tends to refuse a methodology or a structured way of developing a game – the main concern is that it can impair creativity in a field that could not survive without it. A defined ontology would improve and mature the growing industry of digital games, both by improving the understanding of the domain and supporting a structured methodology for designing games. New ontologies improve problem solving within that domain.

This thesis will describe the aspects of digital games and show how they make it difficult to create an ontology for the domain, specially when it comes to a designing perspective. It will follow by disentangling the closest to a unified ontology that there is for the game domain: the Mechanics, Dynamics and Aesthetics framework, i.e. MDA. By reviewing the state of art surrounding this framework, this thesis identified gaps within MDA that impairs its use on a designing perspective, such as the lack of support to some of the described aspects of digital games and how the industry claims that a structured methodology would not support the creativity – a core characteristic of the game's designing process.

To reduce these gaps, changes in MDA's taxonomies are proposed – providing better use for it within a designer's perspective, achieved by embracing the designing aspects of the domain and supported by overcoming issues found in the current literature of game domain.

Keywords

MDA, Game Design, Game Ontology, Game Design Methodology, Game Mechanics, Game Dynamics

Contents

List of Figures

Acronyms

MDA	Mechanics, Dynamics and Aesthetics
GDD	Game Design Document
UML	Unified Modeling Language
RPG	Role Playing Game
MMORPG	Multiplayer Massive Online Role Playing Game
FPS	First Person Shooter
NPC	Non-playable character

Ludology

Tibia	Ciptsoft, 1997
Super Mario World	Nitendo, 1990
Elder Scrolls V Skyrim	Bethesda Game Studios, 2011
Pokemon Yellow	Game Freak, 1999
Cyberpunk 2077	CD Projekt, 2020
League of Legends	Riot Games, 2009
World of Warcraft	Blizzard Entertainment, 2004
Call of Duty Black Ops II	Treyarch, 2012
Far Cry New Dawn	Ubisoft Montreal, 2019
Grand Theft Auto V	Rockstar, 2013
Diablo II	Blizzard, 1997
Diablo III	Blizzard, 2008
Destiny 2	Bungie, 2017
Metal Gear Survive	Konami, 2018
Metal Gear Rising: Revengeance	Konami, 2013
Slender: The Arrival	Blue Isle Studios, 2013

Chapter 1

But Why?

"But Why?" - Ryan Reynolds

1.1 But Why?

This thesis was written by necessity. After being present in two universities studying lectures directly related to games (Game Development in SENAC, Goiania, Brazil and master in Design and Development of Games in UBI, Covilha, Portugal), working as a game programmer in a company with more than 6 years on the market and developing games as hobby for over 10 years, I stumbled across the immaturity of the game domain: there isn't a structured ontology widely accepted by it neither to be used in an academic environment nor to help design and developing of games. Not only because it's a relatively new domain, but also due to some specific aspects that have hardened the creation of an ontology that would support it. The lack of an ontology for the domain decreases the efficiency of game researching - and this inefficiency scales when it comes to designing games. Even basic concepts, such as game Mechanics, which are considered the building blocks of digital games, do not have a fixed and clear definition. Without a defined ontology, trying to propose a structured methodology for designing games is almost naive.

1.2 Objective and Methodology

I do believe that a well defined ontology that supports a designing methodology will enhance the quality of games produced by the industry and also provide an academic support for improving the domain's knowledge. Despite the lack of a unified ontology for the domain, there have been attempts into creating one. By reviewing the current domain's state of art, this thesis was born with a clear purpose to pursue.

This thesis has a bold objective: move one step forward to the direction of a defined ontology that embraces the specific aspects of the domain, while also providing support for it to be used in a designing perspective. The way that this thesis hopes to achieve this is by reviewing the domain's literature on this topic, merging different definitions, describing some contradictions and misinterpretation of even the basic concepts that surrounds the domain and by proposing new taxonomies to support an ontology based on the most accepted framework in the domain: the Mechanics, Dynamics and Aesthetics, or MDA [1]. There are some core aspects that the ontology has to possess to fulfil its objective:

- **Clearly defined taxonomies**, that would not allow misinterpretation or elicit unnecessary complexity in its definitions. To achieve this clarity this thesis will first present the state of art of the current domain's literature, describing the issues surrounding them that goes against this objective and proposing new taxonomies that hopes to overcome this issue;

- **Embrace the domain specific characteristics**. This thesis will first describe these characteristics divided into three categories: the core aspects of digital games, the difficulties that surround developing games and the designing aspects that harshen the creation of a structured design methodology. The proposed taxonomies hopes to explicitly support those characteristics and overcome the designing aspects that impair the creation of a design methodology;

- **Ease the acceptance** by the industry. The proposed ontology is derived from the most accepted framework in game domain: The MDA framework. I hope this fact will shorten the learning curve necessary to understand the proposed changes in its definitions, and by doing so increasing the chance of acceptance by game designers and domain's experts in general;

  - **Useful in a design perspective**, increasing the efficiency of game development processes. This thesis will always focus the use of the proposed ontology within a design perspective, hoping to show that a structured methodology can be adopted without paring creativity - and maybe even enhancing it.

1.3 Thesis Structure

This thesis hopes to move one step towards a definition of an accepted ontology among game designers. In order to accomplish the objective, this thesis starts by first describing some characteristics inherent to games, especially digital games, in the second chapter. The third chapter dives into the difficulties that surround game development. Chapter four follows by describing some characteristics of the domain that harshen the process of developing a structure methodology for design. The next chapter will explain the MDA framework, describing its aspects and why this thesis uses it as a blueprint for the proposed ontology. Chapters six, seven and eight will review the state of art of the basics concepts of MDA: Mechanics, Dynamics and Aesthetics, and finish by proposing changes to support the objective of this thesis. The last chapter will finish this thesis with the conclusion achieved and proposing future works.

Chapter 2

Why Games?

"Playing a game is the voluntary attempt to overcome unnecessary obstacles" - Bernard Suits

2.1 Digital Games

Over the past few decades, the rise of the computer game industry was astonishing. It is no longer made only by individual developers or small teams - it has grown to large companies, involving professionals of different fields, going from programming to sound designer and writer. These teams can easily consist of more than a hundred designers. It has become a multi-million dollar industry. The development of triple-A games has a production cost close to the cost of Hollywood movies, and it can even rival a blockbuster movie. For instance, Grand Theft Auto 5, a game by Rockstar, showed the greatest profitable entertainment product of history - with over 6 billion dollar in revenue. Even for casual games, which could be developed by a small team in a couple months a few years ago, we see these changes in investment and cost of production - and of course, in sales numbers.

2.2 Ludic Devices

Ludic is defined as something that is playful, enjoyable. A digital game is an electronic game that involves interaction with a user interface, with a purpose of being fun, therefore digital games can be defined as Ludic Devices. This is a unique aspect of games when comparing with other entertainment products, such as movies or music. In those products, the user is merely a receiver: is a linear or a one-way connection, not requiring any interaction with it to evoke emotion. Games on the other hand are products that have to be interacted, played by the user. A game for itself does not directly evoke any kind of experience by itself - but it allows the player to invoke them. This implies a core aspect of Digital Games: Games are nothing but a tool, that when interacted allows the user to invoke emotions.

Game as a ludic device will create a set of rules, a world or space where it occurs and the player can interact to create its own experience. The way that the player will interact is not an aspect that the designers have full control of, therefore players can have different outcomes when interacting with it. This implies another core aspect of game, its emergent nature.

2.3 Emergent Nature

Emergence in games refers to the fact that the behaviour of certain games is the result of a complex and dynamic system of rules. The use of the term emergence in games is often in reference to the use of the term within the sciences of complexity, where it refers to behaviour of a system that cannot be derived (directly) from its constituent parts [3]. Emergence is an important source of gameplay and replay value, but it is also very hard to predict, design and control since in digital games the number of possible states is huge: relatively few, and often discrete mechanics can create a large number, sometimes even infinite, of possible states [3].

Due to this characteristic of the domain, hoping to achieve the proposed emotional responses from the players requires a deep understanding of how the parts of the game would interact and be delivered to the final user - and this is far from an easy task, since the nature and emergent behaviour of games is poorly understood [3]. This nature of games as complex rule based systems that exhibit many emergent properties on the one hand, but must deliver a well-designed, natural flowing user experience on the other lies the difficulties when designing games [3], since it will always be too many different players to predict every behaviour that each player might engage in [6].

This is the realm of emergent gameplay, which occurs when interactions between objects in the game world or the player's actions result in a second order of consequences that was not planned, or perhaps even predicted, by the game developers [17]. It is often impossible to accurately predict the behavior of a game before it is implemented [3]. The active substance of these games is not the complexity of individual parts, but the complexity that is the result of the many interactions between the parts [3].

This complexity of digital games nowadays allows virtually infinite outcomes - and one can argue that trying to predict all possible outcomes is naive. Nevertheless, understanding this nature is a must for game designers that hopes to deliver a product that can be used to fill its purposes. By correctly tuning the mechanics of the game, designers will see this emergent nature as an advantage instead of an obstacle: the unexpected possibilities that can emerge from different players' behaviour will mostly support the expected emotional purpose of the game, and not impair it.

Chapter 3

Why Worrying?

Games are hard to design and develop. And it's getting harder by the year - as the technology and industry keeps increasing. In order to keep producing games that can keep the pace in quality standards, companies need to improve the efficiency and effectiveness of the development process. There are a number of guides, methodologies and theories that have been created over the last years to help analysing, designing or documenting games. Some can be used as designing tools, other documentation tools, some are for analysing games, but most of them fail in one aspect or another - and more importantly - they sometimes contradict themselves in describing even the basic concepts contained within the game design domain.

One can find many reasons why this industry is still maturing its development process, and in this chapter i will describe some of them.

3.1 Lack of Ontology

Originating from the philosophical discipline of metaphysics, an ontology represents a structure of entities with the purpose of organizing knowledge and managing complexity in a certain domain. In the Oxford Language Dictionary, it is defined as a set of concepts and categories in a subject area or domain that shows their properties and the relations between them. In computer science, it refers to the identification and (often times formal) description of entities within a domain [5].

An ontology defines a common vocabulary for domain's experts who need to share information. It includes definitions of basic concepts in the domain and relations among

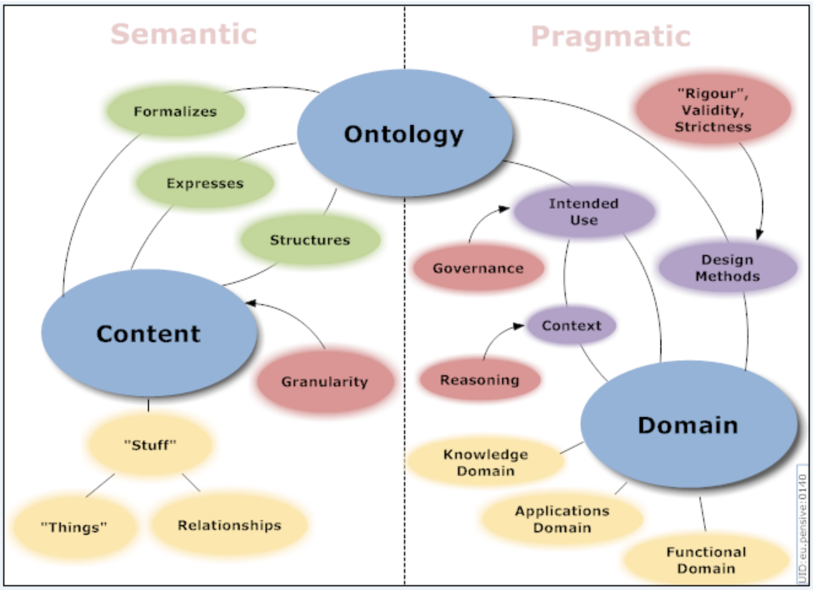


Figure 3.1: The "Dimension map" of ontologies [43]

them. The advantage of developing an ontology for a domain is many, and within the game design domain some of them are:

1. **To share common understanding of the structure of information among the field**, therefore supporting the Game design domain by defining basic concepts that can be used in analysis and research of the field, allowing researchers to share the basic knowledge and concepts, while also supporting the designing perspective by having clear definitions to be used in the processes of developing a game.

2. **To share information among other domains**. The multidisciplinary nature of the game domain merges different domains into one, and a standard ontology can support domain experts to use and share information within their fields with ease.

3. **To enable reuse of domain knowledge**, supporting new research being conducted by providing knowledge and concepts of an existing ontology, that will help both the result of the research as well as the acceptance in the field.

4. **To make domain assumptions explicit**, so the designers can rely less on the previous experience only. Without a defined ontology, the understanding of implicit information is a requirement in the field, and this can impair the efficiency

of designers with little experience on the genre or public target of the game being developed.

5. **To analyse domain knowledge**. A formal analysis is extremely valuable when attempting to extend the domain's knowledge.

There have been some attempts to define an ontology for the domain. A project called The Game Onthology (GOP) [27] is a "wiki-based framework that tries to develop a game ontology that identifies the important structural elements of games and the relationships between them, organizing them hierarchically. The GOP provides a framework for exploring, dissecting and understanding the relationships between different game elements. It is a hierarchy of concepts abstracted from an analysis of many specific games" [10]. Although it was trying to define an ontology for a domain that desperately needs one, the last entry in the GOP project was in 2015, and it seems to be abandoned [27] .

Frank Malcher, André M. M. Neves e Leo Falcão [12] conducted an experiment to test the GOP efficiency in games already in the market. They concluded that GOP, when used, was not capable of modelling with precision various elements found in available games because of the difficulty in specifying non ambiguous definitions for a concept and the enormous number of variations in elements of games and their possible uses. The great number of concepts turned out to be troublesome because it increases the time and the difficulty of the analysis process, the concept memorization process and learning the practical use of these concepts during the analysis process. This is relevant because it is directly related to the effort a game designer must take to be able to benefit from this method [9] [27].

Another attempt was the ambitious "The 400 Rules of Game Design". This project was originated by Hal Barwood and Noah Falstein based on a talk given by Mr. Barwood at a Computer Game Developers Conference [26]. This project had the intent to collect 400 rules of game design but only 112 rules were listed and the last one was added in 2006, and was also abandoned.

The closest that we have from a widely accepted ontology is a framework proposed by Hinick et al: Mechanics, Dynamics and Aesthetics, or MDA [1]. It has been quite influential and frequently used in universities all over the world. Although is somewhat accepted among the field, is mostly used in universities with an analytical purpose, not commonly used by the industry for helping designing games. The main issue that surrounds this ontology found in literature is that its concepts lacks scrutiny and accuracy [9], and can even contradict itself in the definitions [16] [17] [18]. This ontology will be further detailed on the next chapters.

One characteristic that we can see in these attempts, is that they are mostly analytical based. They often do not translate to the real world of designing games, which impairs its acceptance by the industry. The need for an ontology that can work alongside the increasing complexity of designing games is evident. These attempts are also hard to describe the growing genres and types of games. More often than not, these vocabularies are very good at describing particular games, but they rarely transcend into a more generic vocabulary [3], thus reusing these ontologies can be somewhat impaired.

The lack of an ontology within the Game Design domain is a characteristic that impairs the analysis of the field, and even more the designing of games [3]. "Within the game industry, and to a lesser extent within game research too, there is no fixed vocabulary unified vocabulary for describing existing games and thinking through the design of new ones" [5]. "Many concepts are used quite informally, and terminology frequently overlaps or even conflicts" [3]. This issue harshen the growth of the field in many aspects, by both impairing the understanding of the domain from an academic perspective and by leaving the designing process relying upon subjectivity and previous knowledge of its stakeholders. Because of this, there is an urgent need to define one that would increase the understanding of the domain, especially one that can be useful when it comes to a design perspective of the domain.

3.2 Subjective of Fun

Its not an easy task to define what makes a game fun. First, even the concept of fun is loaded with subjectivity. The Oxford Dictionary defines it as "activities that you enjoy", but one can dive deeply into philosophy to encounter different definitions for this term. When it comes to the domain of game design, people would often say that the main objective of a game is to be fun. This idea of an objective that relies on an extremely open and subjective term can bring with it unclarity and uncertainty when designing games. "When writing or talking about games we often hear the phrase “this game is a lot of fun” or the exact opposite. A lot of people have trouble describing why exactly they like or dislike a game, besides of “nice graphics” or “lots of bugs”." [18]

This is one of the reasons why the attempts in defining an ontology for games are mostly based on an analytical perspective. It’s easier to analyse the pieces of a created game then to design a new one: a designing process requires an expected result, an objective. There is not an easy way to describe this objective that won’t flirt with the subjective domain of the human psyche, thus trying to analyse already existing games seems more of an achievable task. Game requirements elaboration is complex, as subjective elements “fun” does not have efficient techniques for its determination [2]. This aspect of the domain haunts the industry since its birth - and it keeps decisions and choices to be made based mostly on previous knowledge of the team involved. These type of decisions are not easy to be communicated or translated to other games genres, becoming an obstacle to a structured designing view of the domain.

Another issue is that without a clear definition of the main objective of games, there is not a compelling reason to adopt a specific methodology for designing it. The subjectiveness of the domain does not ease the task to compare different methodologies: quality is subjective. How can you state that one methodology is better than aother by looking at the end product? You have no choice but to rely on subjectiveness.

3.3 Multidisciplinary Nature

The growth of the industry brought with it an increasing complexity of games. As previously pointed out, game development teams can surpass hundreds of people, and "include practitioners from such diverse backgrounds as art, music, graphics, human factors, psychology, computer science, and engineering. Individuals who, in other circumstances, would be unlikely to interact with each other on a professional basis unite in their economic goal of creating a commercially successful product." [2]

This mixture "seems to produce a true split on the team, dividing it into “the artists” and “the programmers”, a division that basically does not exist in the traditional software industry - and its the main source of important misunderstanding problems, since both teams believe to communicate clearly when using their specific vocabularies to express their ideas" [9]. "This multidisciplinary nature of the video game development process interacting with traditional software development creates complexities that may recommend a specialized software engineering methodology for this domain." [2]

This characteristic of the field that implies a wide range of professionals involved requires a viable and efficient protocol of communication among the stakeholders involved – hence so many attempts on trying to create a standard designing process for the required domain. The communication factor among these professionals is a barrier that must be addressed in order to increase the team's efficiency.

Chapter 4

Why Designing?

"He that would perfect his work must first sharpen his tools" - Confucius

The game industry has recently undergone a transformation more profound than any other in its history. The popularization of mobile games together with the increasing investments in consoles and PC demands games to be developed with more quality than ever in order to keep the pace of the industry - and for it to keep producing better quality for less money, it needs to find ways to either increase revenues, or improve the efficiency and effectiveness of the development process.

Due to the difficulties brought by the aspects of the domain previously described, the attempt of creating a standard methodology for designing games is not an easy task. And as such, has not yet been achieved. The lack of proven, effective design methods is a serious concern. The current design vocabularies for games tend to have a steep learning curve, are not useful for different genres of games and are anything but widespread. "Design accounts for the majority of game development challenges because in many cases it cannot be fully solved or even anticipated at the outset of the development process since the interaction between game elements is unpredictable" [9]. Not only this, there are designers amongst the industry who are reluctant in accepting the idea of a standard methodology or structured process of design [3] due to this emergent nature of games and other aspects of the domain, which will be described in this chapter.

4.1 Second-Order Design

As previously mentioned, the emergent nature of games as ludic devices is one core aspect of the domain. The complexity of games allows it to emerge virtually infinite outcomes depending upon how the user interact with it: interactions between objects in the game world or the player's actions result in consequences that could not have been predicted by the game developers [17] [25], implying that this result cannot be fully controlled [3]. This is often referred to as a second-order design problem: designers do not define the solution, they define something that enables something else create the solution [17] [3] [21].

Design is not just a matter of coming up with a "Great Idea" for a game; it is about coming up with a set of rules that will implement that idea [25]. Emphasizing this indirectness between what the designer puts into the game and the final result of the player's experience can be very useful because one of the most challenging things about game design is how complex and surprising a game's behavior can be. It's easy to add some element to a game expecting one result and then watch something totally different emerge once the game is set in motion and begins to unfold unpredictably through time and space [24].

For game designers this means that understanding the structural characteristics of emergent systems in general, and in their games in particular, is essential knowledge [3]. And of course, not an easy task - seemingly trivial technical decisions in the level of coding and data representation will "trickle upward" and affect gameplay [17]. "In practice there will always be too many different players to predict every behavior that each player might engage in" [1], but this does not mean that designers will have absolute no control over it: a game with an emergent structure often still follows fairly regular patterns [3].

This core aspect of game design implies a barrier when designing it. The challenge to create a game that delivers an expected emotional response or experience is huge due to its emergent nature. The gameplay results that a team can hope to achieve hardly correspond to all that can be invoked by the user within the game's complex rules system. Understanding this concept is important for the domain, specially from a designer perspective. It allows designers to create games where the freedom of the player is balanced with the control of the designer: [...] they do not specify every event in detail before the game is published, though the rules may make certain events very likely. [3]

4.2 Analytical view

The second-order nature of game domain makes it extremely hard to create a structured methodology for design that would work with different genres, since it would have to deal somehow with many unwanted behaviours that may not be expected to emerge when the player is interacting with the game system. This issue added to the subjectiveness of the emotional purpose or objective of the game product - being fun, or enjoyable - creates a consequence for the domain: it's easier to work on analysing already created games then to develop a new one. Most ontologies that surround this domains are based on an analytical point of view. Although many are created for actual design work, few have found frequent use outside universities. This is a general characteristic of game vocabularies. "Their success as an analytical tool does not translate easily to being successful as a design tool" [3].

4.3 Creative Process

No designer would argue that game designing is a deeply creative process. Another common acknowledgement by designers is that they are mostly against the use of structured definitions or patterns within the field, claiming that it would be detrimental to creativity [2][11][3][4]. It is not easy to fit creative or artistic processes in a structured or fixed methodology of work - therefore impairing the acceptance of a structured method by the industry.

The negative effects of patterns and methodologies on the creative process of game design are trickier to deal with. "First, defining patterns creates a risk of viewing them as a methodology for only removing unwanted effects of a design rather than tools to support creative design work" [4]. Second, one must realize that while these patterns seem to be applicable for our use, it should remain clear that not all aspects of design can or should be seen as solving problems, especially in a creative activity which requires not only engineering skills but also art and design competences [4].

4.4 Requirement issue

When developing a digital game, software engineering says that first we must identify the requirements of the system. A task that is unavoidable, and carries some extra barriers in for the game domain: most games aspire to qualities that can be defined as non-functional requirements, such as fun, immersion, enjoyable, rewards and challenge. Non-functional requirements are often called "quality attributes" of a system, or how a system should be, instead of what it should do. These requirements creates special demands on the engineering process within the game domain: "fun" and "enjoyable" are not well understood from the perspective of requirements engineering, causing communication issues between game designers and software personnel [2][11].

The non-functional requirements in game design carries another aspect that makes their identification harder: the subjectiveness of these concepts. As mentioned previously, one cannot precisely define fun or enjoyable - these terms are loaded with philosophical definitions, and hope to clearly define them is naive. They do not have efficient requirement engineering techniques for its determination, thus pursuing them entails building numerous playtests and iterations, usually by programming in an *ad-hoc* manner [11]. And of course, the validation of these requirements fall into subjectiveness due to their nature.

Requirements engineering is difficult – "the ability to precisely communicate and capture stakeholder wants and needs is rare" [2]. The game's domain peculiarities like the lack of an widely accepted ontology, the multidisciplinary aspect of the stakeholders involved, alongside with the subjectivity of its non-functional requirements makes it even more difficult, and clearly identifies the need to extend traditional requirements engineering techniques to support this creative process in video game development [2][11]. This turned out to be a harsh task not only due these aspects, but also how emergent requirements - those that are discovered during prototyping - are a significant aspect of the creative design process, making it difficult to directly transfer general software engineering practices to the game design domain [2].

4.5 Iterative Design

Wikipedia define iterative design as a "methodology based on a cyclic process of prototyping, testing, analysing, and refining a product or process. Based on the results of testing the most recent iteration of a design, changes and refinements are made. This process is intended to ultimately improve the quality and functionality of a design", and most game developers are convinced that this is a critical aspect of developing games [2].

Due to the emergent nature of the game design domain and the unexpectancy brought with it, the gameplay is a result of the game as a dynamic system, and the best way to find out whether or not a game works is to build the system and set it in motion [13]. Game designers typically do this a lot in playtests before they publish their games [22], looking for validation of the previously defined requirements and the identification of emerging ones. These emergent requirements are an important aspect of the creative design process, and only through an iterative process that we can hope to capture them [2].

Iterative production cycles within the game design domain should be a given [6]. The emergent characteristic of the game design domain previously described has a synergetic connection with iterative process - due to the complexity of games the designers can't foresee all the possible outcomes that can emerge from different players behaviour. This complexity implies challenges that cannot be fully solved or even anticipated before set in motion [11], and alongside with the ease of the prototyping burden provided by game engines makes the iterative design process a solution widely accepted by the industry.

4.5.1 Game Engines

Game engine is a software application that includes a programming interface and a number of software libraries that provide high quality graphics and visualization, physics simulation, animation and interaction mechanisms, to aid users in the implementation of digital interactive games for various platforms. They are tools available for game designers to code and plan out a game quickly and easily without building one from the ground up.

With the advent of game engines, the iterative design process has been taken by the game industry as a must. With the facilitation brought by them, added to the emergent nature of games, prototyping every step of the process is a common practice of the industry. Game engines have allowed the designers to create an interactive design environment within the production due to the speed of compiling and emulating the aimed platform, lowering the costs of prototyping and testing the product. The game engines can even allow some of the personnel not directly involved on the software production to work and prototype by themselves: a designer can use the engine and test how new art ideas will affect a scene without requiring software related personal, therefore improving team efficiency by increasing the ability to delegate tasks. Because of these advantages, the iterative design process is in the core of game design domain - a process that is greatly enhanced by the use of game engines.

4.6 Documenting

Most game companies nowadays creates design documents [3] [13]. With a small research, one can find multiple templates adopted by different companies, "although no standard emerged that describes how, when or to what purpose these documents should be written" [3] [2]. Virtually every book that discusses game design has its own template, and give different reasons to create one. Some are in favor of lengthy detailed descriptions covering every detail of a game, while others favor brief documents that capture design targets and design philosophy [2][3].

Many are the reasons that there are not a widely accepted pattern of documentation - and the lack of a standard ontology is certainly one of it, if not the biggest one. Without a structured ontology, added to the creative and emergent nature of games, a game design document is replete with implied information [2]. This consequence makes it hard for it to grow into a standard methodology, since it would require significant domain knowledge, making it hard to be carried over from company to company or from university to professional career [3].

Another aspect that impairs the acceptance from the industry of a standard methodology is the reason to create a document. Some advocates for a pre-production artefact that capture the creativity and main idea of the game [2] [3], containing little information that can be translated to the production phase of the development. Others defend that it can be used to support for production [11], and should be complemented as the game is developed due to the dynamic and iterative nature of game design [13].

The game industry lacks of a documentation pattern [3] [2], but also acknowledges the need for one [2]. Without a widely accepted ontology, trying to create a methodology or standardization for the process of creating documents is nearly an unachievable task. The creation of an explicit ontology would serve to improve the documentation process, and as consequence reduces development cost [9] - and maybe it could be the first step towards the industry's acceptance of a documentation pattern.

Chapter 5

Why MDA?

5.1 Mda

The most widely accepted ontology for the domain is the MDA: Mechanics, Dynamics and Aesthetics [9] [17] [3]. This framework was presented by Robin Hunicke, Marc LeBlanc, Robert Zubek [1] in Game Design and Tuning Workshop at the Game Developers Conference, in San Jose, and it divides the game into 3 elements:

- **Mechanics:** describes the particular components of the game, at the level of data representation and algorithms;

- **Dynamics:** describes the run-time behavior of the mechanics acting on player inputs and each others' outputs over time;

- **Aesthetics:** describes the desirable emotional responses evoked in the player, when he interacts with the game system.

MDA is a framework that supports an iterative approach to designing and tuning a game. By having the desired emotional responses defined, it allows the game designer to reason explicitly about these goals, identify Dynamics to support them and define the Mechanics accordingly.

The whole point of MDA is to help guide designers through game design's highly second-order creative process [24], addressing how changes in each layer might affect the game as a whole [3]. The MDA framework proposes that designing games is about understanding that a designer only has "direct control of the game's Mechanics; the Mechanics work together to generate the Dynamics, which in turn generate the Aesthetics" [17]. In other words, two-thirds of the final product (dynamics and aesthetics) are not under direct control [25] - this is the challenge contained within the realm of second-order design. MDA offers a way of thinking about this challenge – it develops models that help illustrate all the different kinds of qualities of player experience you might want to generate, the different kinds of dynamics that might produce those experiential qualities, and the types of mechanics that are likely to lead to those dynamics [24].

"The MDA framework in an attempt to bridge the gap between game design and development, game criticism, and technical game research, while strengthening the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts" [1].

5.2 Problems

Although is the most known ontology for the game domain [3], this framework shares contradicting reviews among game designers and researchers of the domain: some argue that it is effective and can be used for both designing and analysing of existing games, while others argue that it mostly an analytical tool [17] [16], therefore translating from the analysis to designing of game is not always effective. Another common issue that surrounds this ontology found in literature is that its concepts lacks scrutiny and accuracy [9], and can even contradict itself in the definitions [16] [17] [18], specially a misinterpretation of Mechanics and Dynamics. Some of the reviews about this framework also points to how it can leave some of digital game's pieces outside the definition of what can impact the invoked Aesthetics - such as narrative or story telling for example [6].

5.3 But Why?

Despite these problems, it has contributed for maturing the field. Its an ontology that embraces the emergent nature of games when proposing a second-order of design: "Mechanics is used to refer to the parts of the game that the designer has direct control over, aesthetics refers to the qualities of player experience that the game ultimately generates, and in between, linking the two, are the dynamics of the game in action – the behavior of the game's different parts interacting with each other and the player while the game is being played" [24] (see Fig 5.1). Also, by acknowledging the emergent nature of games and as consequences the existing of emergent requirements, it creates an incentive for the industry to adopt the iterative design process, since only by prototyping each iteration one can hope to identify these requirements [2] [11].

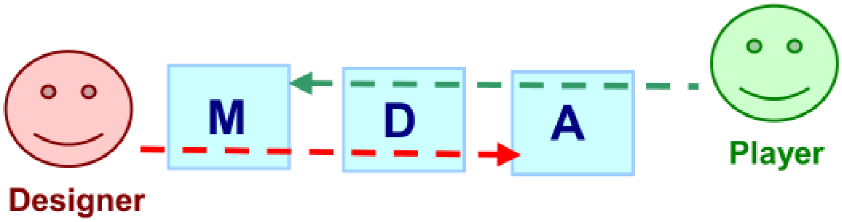


Figure 5.1: MDA [1]

In my opinion, MDA is a framework that differs from others because of one important aspect: it brings the expected emotional response as a core concept to be worked with. Since games are a tool to invoke emotion, this is the main reason that makes MDA the closest to a formal ontology for the game design domain. MDA can be seen as a guide of how designing games should be executed: by having always the emotional purpose of every iteration step on focus, it will help the team to stir the development and make decisions in a more objective way, without relying solely on the designers previous knowledge of the domain.

The MDA framework will always put on perspective the emotional purpose, and with the advantage of not limiting it - but to give a blueprint of how to introduce the main objectives within every step of the process. It acknowledges the subjectiveness of the emotional objectives, and while it proposes eight different emotions [20] called Aesthetics, the framework makes it clear that more might exist and can be worked with [1]. Acknowledging this allows the framework to be used in companies that has a deeper understanding of its users, as it can embrace a more detailed division of Aesthetics with ease.

Another aspect of this framework that makes it easier to be adopted by the industry is that it does not contain a explicitly definition for a pattern or structured way of creating documents, and it can be used alongside pre-existent documentation techniques, such as Game Design Documents (GDD) or even Unified Modelling Language (UML).

I believe that the domain can use the MDA framework as a starting point, but there are gaps in it that have to be addressed. The unclarity in its concepts are a major concern, and is one of the main reasons why this framework is not widely used among professionals while developing games. The next chapters of this thesis will break down the concepts of the MDA framework, comparing it with other taxonomies found in literature, pointing out issues that surround it and proposing changes to address these issues, especially through a design perspective.

Chapter 6

Mechanics

What are game mechanics? This is a widely discussed topic in Game Design. Game mechanics are often considered to be the building blocks of games, therefore an important concept within this field. Based on its importance, many authors have defined mechanics before, and although we can sometimes find similarity between some of the definitions found in literature, this is not always the case. Some authors agree that we have a common understanding of what mechanics is [14], while others concluded that these definitions are neither precise nor conclusive [16].

This chapter will dive into the literature of game mechanics, analysing some of its definitions and pointing some advantages and disadvantages in each one of them. It will then propose a definition that supports this study, and show some examples of how it can be used while designing games.

6.1 State of Art

In literature we can find some statements that fail to propose a coherent definition for mechanics, either by being weakly defined – "Mechanics are the things provided by the game." [18] – or by eliciting unnecessary complexity – "[...] splits game mechanics into 38 separate categories but also holds that more might exist." [14] (see Fig. 6.1). Both of these cases are not optimal for the industry's acceptance of an ontology.

The first definition goes against this thesis's objective of defining a ontology the game design domain because, as it was previously mentioned, an ontology must have its concepts well established, and a weakly definition that allows it to be interpreted

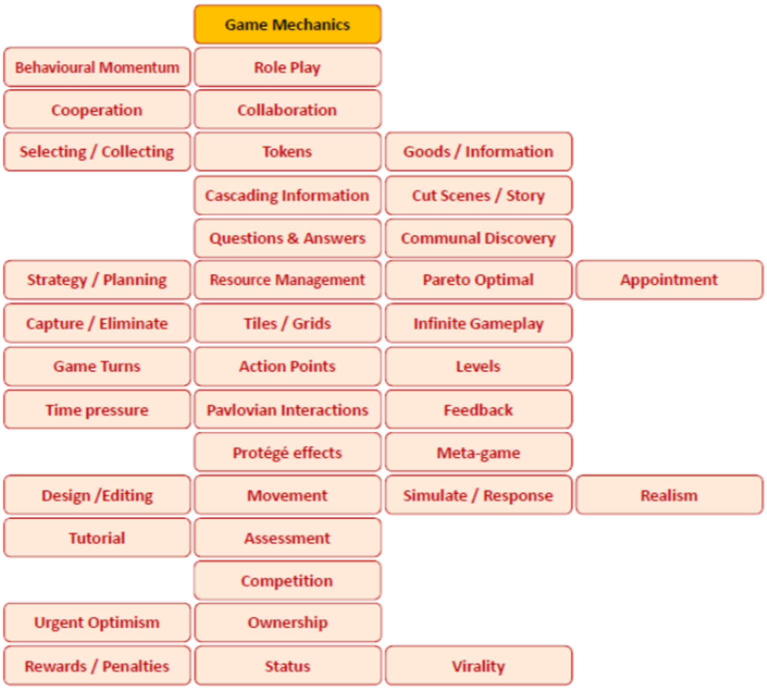


Figure 6.1: Mechanic division [14]

in different ways is not optimal and would not increase the acceptance of this ontology by the industry. In order to increase the acceptance chance, a standardization becomes necessary - hence the necessity of clarity in the definitions of these concepts contained in the game ontology. For instance, with this definition, the language setting of a game can be considered a mechanic, or any other setting like volume or screen bright, and as it will be shown later in this chapter, these should not be considered as such.

The problems that surround the latter definition is that it goes against one aspect that would ease the acceptance of this ontology by the industry: its simplicity, or its capability of being understood and adopted by stakeholders on different fields within the Game Design Domain. By bringing unnecessary complexity to this taxonomy, unwanted consequences will arise: First, the necessity of a more extensive knowledge of the domain *a priori* to understand this concept; Second, the time spent during design to work around finding and defining mechanics would increase; and Third, it would certainly impair the industry's acceptance of the ontology.

In the MDA paper [1], the authors presented us with the following definition: "describes the particular components of the game, at the level of data representation and algorithms". The problem with this definition is that it is once again weakly defined

- one can understand that pretty much all related to digital games can be defined as mechanics. The sound, user interface, network data, platform dependent codes, even game engines, all would be considered components of the game, therefore defined as game mechanics - and once again - this is not an optimal way of describing mechanics to support the creation of an ontology. What would be the point to work on defining mechanics since most of the game can be considered one? It would certainly have a high cost for the team to define them all, and more so to work with each one of them.

Later in the paper, the authors described mechanics as "[..] the various actions, behaviors and control mechanisms afforded to the player within a game context. Together with the game's content (levels, assets and so on) the mechanics support overall gameplay dynamics." [1]. As it can be noticed, the two definitions contradict themselves. The latter one pointed out that the mechanics work alongside - and are therefore distinct from - the game's content, which would be considered mechanics in the first definition. This unclarity of the concept that can be considered the building block of digital games is a common complain around the MDA framework, as Sicart [16] agreed by pointing out that "The MDA framework [...] is a model that does not allow for the description and analysis of a mechanic due to a relative inconsistency in the formulation of the definition.".

In the book The Art of Game Design [7], the author starts with an open definition of mechanics: “the interactions and relationships that remain when all of the aesthetics, technology, and story are stripped away.”. It's important to point out that he uses the term aesthetic here as the art or visual style of the game, not as the player's emotional response as the MDA framework does. This definition can avoid some problems that were previously shown here - such as algorithms, UI or maybe narrative being mistakenly interpreted as mechanics - but it is still a vague definition, too open for interpretation. We can understand that the player's ability to run, jump, crouch or fly is a mechanic - but we can also fit the player itself as mechanics. Or the speed that the player runs. There is no distinction between the objects of the game, their actions or the set of rules that governs it. By using this taxonomy, the number of game's components that can fit in this category would elicit unnecessary complexity within this concept and cause it to not be useful when designing games, although it surely can help to analyse them, as acknowledged by the author himself: “Some authors have approached this problem [defining mechanics] from a very academic perspective, more concerned with an analysis that is philosophically watertight than with one that might be useful to designers” [7].

The book follows by disentangling mechanics into six different categories: Space, Objects, attributes and states, Actions, Rules, Skills and Chance. Added to the problem of complexity caused by the amount of aspects that fall into this definition, the author introduces skill as a concept merged with the definition of mechanics, and he breaks it down into three parts: Physical skills, Mental skills and Social skills. Apart from once again increasing complexity, the blending of the player’s perspective (Aesthetics) with designers perspective (Mechanics) goes against the core idea of MDA proposed by its authors: that the player and the designer view the game from an opposite side of the MDA framework. This blending is even clearer when the author writes: "The mechanic of skill shifts the focus away from the game and onto the player". Within the MDA framework, Mechanics is something that designers have control of - and we clearly do not have control of the players skills. We can only account for the skills that the player is supposed to have and tune our mechanics to achieve the wanted Aesthetic purposes.

Sicart [16] presented us with the following definition: "game mechanics are methods invoked by agents, designed for interaction with the game state". Although this taxonomy is strongly defined and should allow correct identification of Mechanics by explicitly stating how they are invoked, it lacks on supporting its purpose to evoke Dynamics: saying that it is designed for interaction with the game state is not the same as saying it has a defined objective. This taxonomy can also create a wrong correlation of Mechanics and an execution environment, i.e. run-time behaviour, which is a characteristic afforded to Dynamics, and will be further detailed in the next chapter.

There are plenty more definitions for game Mechanics in the current domain’s literature, but the point here has been made. There is not a definition widely accepted, and the lack of precision in this taxonomy is recurrent, as is the lack of support to a designer perspective of the domain.

6.2 The Proposal

As we can see, there are many definitions for mechanics in Game Design. This lack of conceptual precision points to a definitional problem: it is unclear what game mechanics are [16], and this fact would not support neither the definition nor the industry’s acceptance of a game domain ontology. Hoping to overcome this issue, the following definition is proposed:

"Doing responsibilities of Entities, with a purpose to invoke Dynamics"

In the Oriented Object Paradigm (OOP), an entity is defined as any singular, identifiable and separate abstraction of an object. It mainly has two responsibilities: *knowing* and *doing*. *Doing* are the actions afforded to an entity, while *knowing* is the information it possesses. Within the game domain, in a hypothetical first-player-shooter game, the player, a gun, an enemy, the map, all these are easy identifiable entities, since they are some of the core objects of the game. Other components of the game are also entities, but harder to define since they belong to a lower abstraction level, such as the game camera, gravity, or the UI interface.

The first part of the definition avoids that entities would be mistakenly defined as Mechanic. In the previous FPS example, the player, a gun, an enemy, and all the entities described would not be defined as Mechanic. If we use the well known game Super Mario World as an example, the player is an entity that can jump - an action, or a *doing* responsibility - and also knows its current move speed, or its current position in the world - *knowing* responsibility. The *knowing* responsibilities would already be discarded and not defined as mechanics, while the doing responsibility needs to go through one more step to be defined as Mechanic: check its purpose.

The second part of this definition makes it clear that it has to have a purpose: to invoke Dynamics. This is important to avoid many unwanted actions of the game's entities being considered as Mechanics, such as a game camera controlling the aspect ratio, or the UI responding to a pointer click. These *doing* responsibilities do not have a direct purpose of invoking any kind of Dynamic - they are just necessary to allow the user to interact with the game.

Is it important to note that although this definition inherits some of its terminology from the Object Oriented Paradigm, it does not claim that it requires any previous knowledge in software engineering to understand and use it. With this definition, this thesis hopes to achieve the following objectives:

- **Strongly Defined Taxonomy** The definition will support a clear ontology by avoiding a narrow concept that is open for interpretation in a way that would cover unwanted aspects of Game Design as Mechanics. We approach this by trying to explicitly specify the responsibility afforded to an entity - *Doing* and by making it clear that this action must have a purpose: invoke Dynamics. With this definition, a entity by itself should not be mistakenly considered mechanics (e.g. a player, a gun, a platform), but its respective responsibilities (e.g. run, shoot, move) should be analysed, and if it contains a purpose to invoke Dynamics, it should be defined as a mechanic. This will avoid a definition that would allow unwanted aspects of the game to be defined as mechanics - such as language and volume settings, or network algorithm - without eliciting unnecessary levels of complexity that would be counter productive for the design process.

- **A functional correlation - or *flow* - between Mechanics-Dynamics-Aesthetics** By specifying that a Mechanic has to have a purpose to invoke Dynamics, this definition ties itself to the other end of the MDA framework (see Fig 5.1), creating a connection with the Aesthetic response expected trough Dynamics and specifying the direction of influence in this relationship, acknowledging the second-order of design.

- **Support a Design Perspective** By explicitly defining that it is a responsibility of an entity, this taxonomy would not cause insane amounts of pointless Mechanics to be mistakenly elicited in the development process, reducing the time cost and error chance of this step, thus making it a more efficient process to be used in designing. By defining its purpose, this taxonomy clarify the reason why each Mechanic is present in the game - to invoke Dynamics that ultimately will reach the player's emotions (Aesthetics), allowing the team to correctly tune the game's Mechanics when needed and understand how each Mechanic would impact the player experience of the game, and giving a strong reason for Mechanics to be defined in the designing process.

- **Ease Communication** By inheriting concepts of the OOP, the encapsulation of unnecessary complexity will ease its utilization by professionals of different fields involved in game development. When the purpose to invoke Dynamics is defined, it will allows that every Mechanic elicited to be explained in an abstract level of a game itself, not worrying about platform dependent code, software or design knowledge, making it easier to be understood by different professionals, supporting the multidisciplinary aspect of the domain. For instance, a player's Mechanic of shooting, that invokes the Dynamic of killing enemies, can easily be understood by the team involved in the development. If the proposed definition allowed *doing* responsibilities of entities that do not have a purpose of invoking

Dynamics to be considered Mechanics, action such as "controlling aspect ratio" afforded to the camera, or "check for collision" afforded to enemies would be considered Mechanics - both of this actions requires specific domain knowledge to be understood, and as such could not be an easy task by all the stakeholders to understand it, impairing the communication among the team.

- **Documenting** The borrowed entity concept from OOP brings a perk to the domain when it comes to the documenting process: it can be used synergistically with standard documentation techniques already adopted by the software industry, such as the Unified Modified Language (UML). This pattern can be used in a higher level of abstraction, both formal and informal ways, increasing the communication efficiency of the team as well.

With this definition, we hope to give directions when designing a game on how to find its mechanics, by identifying entities and its doing responsibilities related to the emotional expected response of the player.

6.3 In Use

The importance of defining Mechanics is clear: it is what the designer can directly control in order to achieve the wanted emotional objectives, or Aesthetics. Identifying them is at the core of designing a game, and should neither be overlooked nor cost more time than necessary. Although its the first layer that connects designers to players, its the last layer to work when starting the development process. Since Mechanics have the purpose to emerge Dynamics, which will then invoke Aesthetics, we must first have the objectives in mind: define the wanted Aesthetics, the Dynamics that would invoke them and only after define which mechanics would be necessary to allow these Dynamics to emerge. How to find and define Dynamics and Aesthetics will be described in the next chapters.

- **Implied Mechanics** The development team must be aware of implied Mechanics. These are the doing responsibility of entities that should be carefully analysed to be defined as Mechanic. In a regular platformer 2D game as an example, the player can walk, jump, die and get hit. If all these kind of actions were to be defined as Mechanics, the purpose of avoiding huge amounts of defined

Mechanics would not be fulfilled. We have to deal with these kind in a simple way: if it can be tuned to **explicitly** invoke Dynamics, then it should be defined as Mechanics. The action of jump for example: it can be ignored in a hyper-casual platform game if its only purpose is to jump obstacles, since there is nothing to be worked around it to directly invoke any Dynamics. But if it is a more complex platform game, where Challenge is one of the Aesthetic priorities, the player can have a skill of "double-jump" that could directly create a Dynamic of escaping some enemies attacks by using it, or trying to double-jump in a correct time and place to overcome obstacles, and in this case, it should be defined Mechanics and worked with.

Consider as an example a RPG single player game, where the player is supposed to be immersed in a huge and well-detailed fantasy world. In this case, Discovery (game as uncharted territory) is picked as the main Aesthetic. In order to enhance this Aesthetics, some Dynamics can be defined, such as the Dynamic of exploring the map to find hidden chests with good rewards, to create an incentive for the player to chart new territory. By having defined this Dynamic, we need to identify the entities that have Mechanics to invoke it. One Mechanic would be that the player's avatar can open the reward chest. This Mechanic can be considered implied for those who are familiar to the genre and sometimes not directly involved in any Dynamic, but in this case it can explicitly support the Dynamic of looking for secret places. By defining it, the team can work on how to tune it in order to enhance the expected Dynamic: they can work on sensorial feedback such as a well detailed animation and sound when finding or opening the chest, or creating a prize or reward after finding a predetermined number of chest - both of these tuning would increase the chance of the user to fulfil the Dynamic that supports the wanted Aesthetics.

Another example is gravity. Is it to be considered a Mechanic? In games that simulate the real world, probably not, since its purpose is implied as a must, and usually encapsulated by the game engine's physics motor. Now if we think of a space game where we want to invoke Challenge by creating a Dynamic of landing a spaceship in different worlds to progress the game, where the gravity would be different in each one of them, we should define Gravity as a Mechanic of the game world.

What is a mechanic attached with a gold coin? It knows information, like its value in points, its position in the game world, its graphical image. And it can do actions, like animate, spawn, or be destroyed when the player takes it. Which of these would be considered a mechanic? It depends. If the sole purpose is to give points to the player, it is just an entity without an associated Mechanic. The Mechanic here can afforded to the player - take coins - unless it has some specific function that could be worked to enhance some Aesthetic, such as teleport to a different location after some time, to create a Dynamic that the player must try to pick it as fast as possible, or it will teleport and the chance will be missed. This Dynamic would invoke Challenge Aesthetic.

- **Core Mechanics** In a FPS game the player has a shooting Mechanic. In a RPG game the player has the Mechanic of attacking enemies. Both are considered core Mechanics: they are implied ones and also represent the most importance in invoking Dynamics, and as such they should always be defined. A team can work towards any prioritized Aesthetic through the core Mechanics.

  In the FPS example, to enhance Challenge Aesthetics the team can work on tune the engaging in combat Dynamic, increasing the delay time reloading Mechanic to increase difficulty, or creating sensorial feedbacks (text on screen or sounds) when killing multiple enemies in sequence as reward for overcome a Challenge. If Fantasy is an Aesthetic priority the team could work on removing the "floating texts" from the Mechanic of attacking enemies that shows how much damage they suffered from shooting to enhance immersion by being similar to reality and create a more believable game world. If its Sensorial a prioritized Aesthetics, the team could work on creating multiple animations for enemies getting shot, and/or increase the detail in these animations, as we can se in Fig. 6.2

  A team should devote more time in balancing the mathematical progression in enemies attributes (such as attack damage, defence and health point) in the RPG example if Challenge was a priority, in order to tune the Mechanic of attacking enemies to a point where the killing enemies Dynamic is not extremely easy, but not too hard to a point of being impossible - just the correct amount to enhance the hunting Dynamic that can invoke Challenge Aesthetic. If this game has Fellowship as a prioritized Aesthetics, the development team should work on creating enemies that would be better confronted by a team of users, and/or give better rewards if defeated by more than one player, creating an incentive

Figure 6.2: Sniper Elite 4: killing animation [29]

for invoking a partying hunting Dynamic.

- **Extra Mechanics** These are the Mechanics that are usually defined later, sometimes after prototyping. This is the category of Mechanics that one can say that they are the "extra" of a game. The extra Mechanics will give the difference between similar genre games. For example, in a puzzle game where the user has to move together blocks with the same colour to destroy them and progress to the next level, Challenge is the main priority, the game camera is an entity. It is implied that the camera has to be there and it has functions and responsibilities like controlling aspect ratio and screen size, and they are to be treated on the lowest scope of software development only, since does not have an Aesthetic objective clearly affected by it - the camera has to exists only to allow the game to work. In this case the camera should not have a defined Mechanic in this game. But if the team has time and resources to work more on that puzzle game, they can create Mechanics afforded to that camera that would support some Dynamics and enhance the chance of invoking the Challenge Aesthetics: the camera can shake when the user is playing at a harder level and destroyed blocks of the same colour 3 times in a row for example, and this would make the game harder. This will create a Dynamic where the player would need to keep some blocks of all colours without destroying as a safe measure to avoid the shaking camera - a difficult task - and will certainly enhance the Challenge Aesthetic of the game.

  In a hypothetical first-person terror game, the Sensation Aesthetic is an important feature to elicit fear. Of many possible Dynamics that could create fear, we can think of one where the user needs to run away from a ghost in a haunted forest that resembles terror movies, while being careful to not get lost - and then find even more ghosts. There are plenty of entities that could have Mechanics to invoke this Dynamic, and the game camera is one of them. The team could work on the camera's Mechanics like creating fog or blur effects, shaking if the ghost is near, or darken to increase fear even more. All these Mechanics are not implied nor core ones - but it can surely enhance the quality of the final game.

There are some Mechanics really specific to one game, and these usually are not considered Mechanics to most others. The User Interface (UI) is usually an entity that do not directly create any Dynamics, and as such hardly contains defined Mechanics. But we can think of a RPG game that Expression is one priority, and to achieve that the team could create Dynamics such as allowing the user to choose among a variety of paths in magic to progress in different ways, equip the avatar with well-detailed coloured equipments and also choose skills that creates detailed visual feedback when used. In this game, the team could create a Mechanic afforded to the UI to change its visuals based on the path chosen by the player: it can turn red if he chooses the fire path, or blue if it was ice for example. This can emerge a Dynamic where the player would try to pick a path, equip some coloured equipment and skills to make them all blend together, in a way of expressing himself. This example showed how one Mechanic can be created in an unusual entity to emerge Dynamics that can help to invoke the wanted Aesthetic.

Imagine a multiplayer first-person shooter game, where friends can join forces against enemies (NPC's), and Fellowship is a priority. The map of this game is quite big - and sometimes friends can move apart from the team, and even get lost. This issue will for sure impair Fellowship to an extent: since players would have to spend time looking for their partners in order to regroup, the Dynamic of killing enemies with your friends (one of the Dynamics that would invoke Fellowship) will be impacted. In this case, an unwanted Dynamic would emerge: players would have to stop killing enemies to search for their friends. This Dynamic should be removed, and this will be further detailed in the next chapter. To solve this, the developers could create a new Extra Mechanic afforded to the players entity: the avatar would have the skill to instant teleport to the group leader. This simple Mechanic would remove the unwanted Dynamic and at the same time enhance the Dynamics that invoke Fellowship.

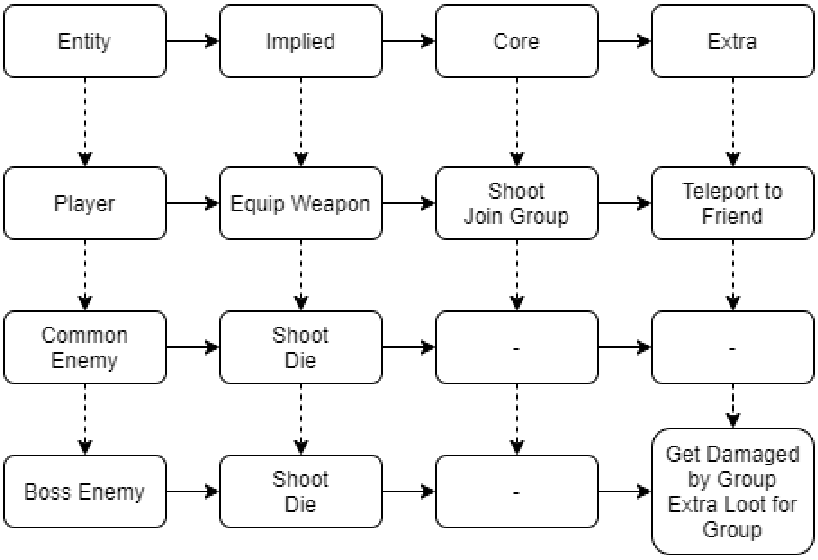
In the same game, we could create a Dynamic that would also enhance Fellowship: make players hunt and kill boss monsters together, as a group. There are several Mechanics to invoke this Dynamic - and an example would be an Extra Mechanic afforded to the boss enemy, Get Damaged by Group. The boss would receive extra damage when attacked simultaneously by all members of a group, making it easier to kill. Another example is an Extra Mechanic that would invoke the group hunting Dynamic by making the boss give extra rewards to all players if defeated by a group, create an incentive for them to hunt as a group (Fig. 6.3).

6.4 Summary

Mechanics are the only thing that designers have full control when trying to make the game fulfil its emotional purpose. Understanding how they work to create Dynamics is necessary knowledge: It shows where the team should spend time and resource working and how it will affect the players experience when interacting with the game. By defining Mechanics, the team will know where to work first (core Mechanics), and how they can enhance the game by creating extra Mechanics that would help to invoke the wanted expected Aesthetic on the player's side, as we can see on Fig. 6.3, a diagram that shows entities and Mechanics present in the last game example cited above.

As we can see, the Mechanics division (Implied, Core, Extra) can help developers to better understand the reasons behind all Mechanics afforded to an Entity (Player, Common Enemy, Boss Enemy), and the order that they are usually detected. If the team wants to create a new Entity that has a connection to the group hunting Dynamic (e.g. a specific weapon that is stronger when used by many players of a group), it would be added to the diagram and become easier to map its influence over the expected Dynamic. This is a simple example that shows how Mechanics works to create an expected Dynamic, and how developers can easily map them among the proposed categories.

This taxonomy was defined based on a design perspective - hence the tight specification. It tries to avoid a complexity that would increase the cost in production,



Entity	Implied	Core	Extra
Player	Equip Weapon	Shoot Join Group	Teleport to Friend
Common Enemy	Shoot Die	-	-
Boss Enemy	Shoot Die	-	Get Damaged by Group Extra Loot for Group

Figure 6.3: Mechanics that invoke "group hunt" Dynamic

but also not being so open for interpretation that its use of it would not be justified. By following this definition, we believe that the team could increase its efficiency by always having the emotional purpose in mind and understanding what they need to create or tune to achieve that purpose.

Chapter 7

Dynamics

> *"Game dynamics are the next level of complexity, and describing them is much harder. [...] they are almost impossible to capture in text form. [...] actual dynamic often remains elusive. [19]"*

The D in MDA stands for Dynamics. This chapter starts by analysing definitions found in literature, elucidating the importance of this concept and its major aspects in addition to the difficulties that rely upon designing games when it comes to defining it. Based on these aspects, it ends by proposing a taxonomy to support this study - and presenting some examples of how it could be used in real games.

7.1 State of Art

What are dynamics? This is a concept that once again one can find different definitions within the literature, although they share some major aspects of this taxonomy. The MDA authors [1] defines Dynamic as "the runtime behavior of the Mechanics acting on player inputs and each other's outputs over time", and from this definition we can withdraw two main aspects of this concept: Its runtime behaviour and its close relationship with Mechanic - a relationship that has a direction of influence from one to another: Mechanics create Dynamics. The first aspect makes it clear that it happens during the interaction of the player within the game, and the latter aspect is acknowledging one major characteristic of the game domain: its emergent nature.

The first aspect is corroborated multiple definitions found in literature, such as "A game dynamic is a pattern of loops that turns them into a large sequence of play" [19] and "[Dynamic is] the behavior of the game as a system – what happens when you play" [28]. Both clearly support the runtime aspect of Dynamics as defined in the MDA paper, or how it emerges in an execution environment - when the game is being played.

The second aspect is also acknowledged in many definitions found in literature: "Dynamics emerge from mechanics" [20] and "Dynamics are the result of the interaction between the player and a Mechanic" [8]. This aspect explains a previously described characteristic of game design that designers should be aware of: the second-order of design. This characteristic tells the designer how he should approach the task of creating Dynamics: by creating or tuning Mechanics. As Mathew Galant stated, "Game designers only have direct control of the game's mechanics; the mechanics work together to generate the dynamics [...] They want to make their games fun and engaging, but only have indirect control of the player's experience" [21]. In these definitions, Dynamic can be seen as the bridge that connects the designer with the game (Mechanics) and that the player with it (Aesthetics) [1] [18].

It is clear that exists a connection between Mechanics and Dynamics - but from this relationship emerges a problem: the misinterpretation of one by another, as stated by Le blanc: "There's a gray area between the notions of mechanics and dynamics" [20] and corroborated by Kritz et.al. : "As players interact with mechanics through the games, it is actually easy to believe they will be confused about what is mechanics and what is dynamics. Dynamics are the result of the interaction between the player and a mechanic, it is only natural to confuse them and take one for another." [8]. The source of this misinterpretation is due to the complexity merged with this concept: the emergent nature of this domain has as a consequence unpredictable behaviours that emerge from the player's interaction with the game, and because of this, the number of possible Dynamics are virtually infinite. This is the realm of emergent gameplay, which occurs when interactions between Mechanics and player's actions result in a second order of consequences that was not planned, or perhaps even predicted, by the game developers [17] [1]. Game dynamics is all about predicting [28], although "in practice there will always be too many different players to predict every behavior that each player might engage in - Dynamics always adds a layer of emergence and unpredictability" [6].

Dynamics is what connects players and game designers [1], it sets the "tone" of the game [19]. The importance of a coherent definition for Dynamics is evident - but not yet achieved by the game domain, especially when it comes to a definition that can be used in the designing process: due to its complexity and virtually infinite outcomes, there is no apparent benefit for the team to waste time trying to find them. This concept is mostly used in an analytical perspective: analysing already existing games and Dynamics seems a more achievable task.

From the literature we can withdraw four aspects that belongs to the Dynamic taxonomy:

- **Its run time behaviour;**
- **The close relationship between Mechanics and Dynamics, and its direction of influence;**
- **The unpredictable nature of Dynamics;**
- **The complexity of Dynamics**

The definitions found in literature fails to create a way of dealing with all these aspects when it comes to a design perspective. They contain a gap when it comes to creating a solution around the unpredictable nature of dynamics and its complexity, therefore relying on these definitions do not ease the process of finding the Dynamics of a game during the design. Since the proposed ontology of this study is based on a design perspective, these definitions do not fully support it.

7.2 The Proposal

This study proposes new definitions to the MDA framework in pursuit of a structured ontology for the domain, one that would not only be used in an analytical perspective, but one that could be translated to the real game design world. The lack of a precise and defined taxonomy does not support this objective. How can a development team work on achieving the Aesthetic purposes of a game if they do not explicitly work with the Dynamics that invokes them? There are two common ways the industry deals with this: by relying on previous similar game's Dynamics and/or by prototyping it in an *ad hoc* manner until wanted Aesthetics are somehow emerged from it.

The first way is more common, and it brings as consequence to game designing the impairment of creativity: the team usually do not have a clear knowledge of how Dynamics work together to invoke the wanted emotional purpose (Aesthetics), so as a risk avoidance technique they rely on copying Dynamics from previous similar games, specially if the team does not fully understand how it emerges, or how it works on invoking Aesthetics. This is not a healthy process for the domain, as it can lead to a evident game's resemblance between different titles. How many RPG's does not have the Dynamic of hunting monsters for in-game currency rewards (e.g. gold coins)? This Dynamic is so present in this genre that is hard to imagine games without it, but they do exist. A game called Pokemon for Game Boy was a huge success and created another Dynamic that would incentive the players to hunt monsters: they could capture them and the player's captured monsters could increase its level by earning experience (see Fig. 7.1).

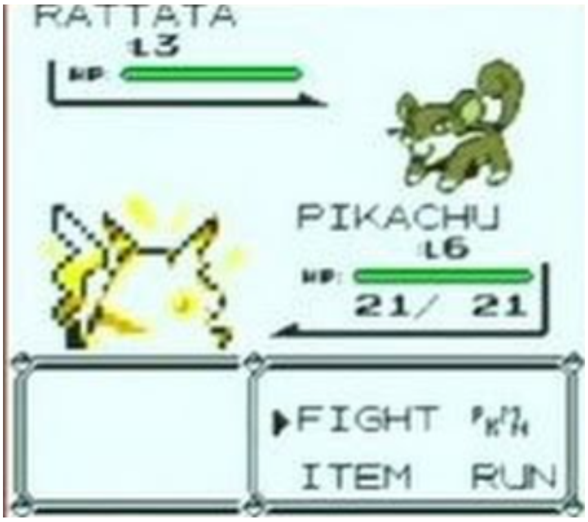


Figure 7.1: A battle in Pokemon Yellow [40]

The second way is an inefficient way of working with Dynamics for a game: although there are some useful Dynamics that emerges only in prototyping and play-testing, without a clear understanding of how Mechanics work to create them the team will hardly be able to invoke all Dynamics that could be created to enhance the wanted Aesthetic, and it will certainly be more time costly. If the taxonomy was well defined and understood, the development team could intendedly work on specific Mechanics to create expected Dynamics, without relying on "luck" and hoping for it to emerge during play-testing. And of course, there is a greater risk of unwanted Dynamics to be shipped in the released game.

To illustrate these ways, let's imagine two development teams working on creating a classic RPG game. The first one understands how Dynamics works, the second does not. After some understating of the public target, both teams focus on achieving the Explore Aesthetic. The first team will think of Dynamics that could enhance this Aesthetic at early stages of the designing process, such as creating ways of "fast travelling" to discovered places - so the players will explore the map in order to find these places and later on be easier to go to more distant places (we can see that in 7.2), or create "mounts" that walks faster than the player (see Fig. 7.3) - easing the exploring Dynamic. The second team does not understand how Dynamics invoke Aesthetics, and as such they could try to improve the visual aspects of the world hoping that this is enough to make the player explore uncharted areas. The second team would eventually understand (after play-testing) that they need to create incentives such as the ones created by the first team, but since it required more iterations, will surely be more costly.

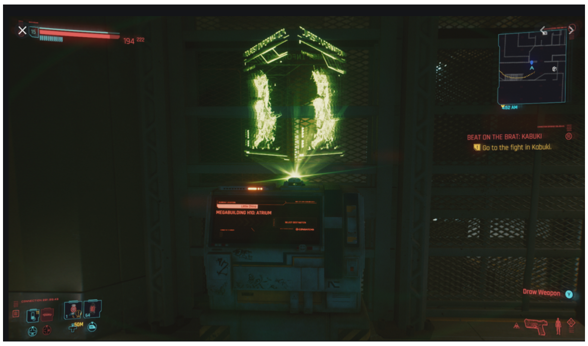


Figure 7.2: Cyberpunk 2077 fast travel device [41]

Figure 7.3: World of Warcraft flying mount[38]

Since its a classic RPG, the Dynamics of quests will be present. When the first team starts to work on creating them, they would understand how to better tune this Dynamic to achieve the wanted Aesthetics: they could create quests that requires the player to travel to distant places, or they could even create a new Dynamic of "finding secret quests", by adding quest starting points to hidden places in the map and giving extra rewards to players that finishes them, and by doing so creating another incentive for the player to explore. The second team could for example rely on previously existing quest Dynamics that can work for other games, such as creating quests full of detailed narrative and massive dialogues, or quests that contain a really hard boss at the end. Both of these quests styles can certainly work in many games, those that have Narrative or Challenge as wanted Aesthetics, but they do not support the achievement of the wanted Aesthetic - and they can also fail when it comes to the public target of these games: those who are looking for Exploring Aesthetic.

These examples showed how understanding Dynamics can enhance the development process, by increasing its efficiency and the quality of the final product. And this is why the domain needs a taxonomy that is well defined and correctly absorbs the four aspects of Dynamic. This study hopes to achieve it by proposing the following definition for Dynamics:

"Dynamics are the predictable runtime behaviours that emerge from Mechanics, with a purpose to invoke Aesthetics."

It's important to note that since this is a design based taxonomy in a domain that has absorbed the iterative process, it should be considered as predictable the Dynamics found in any iteration step of the process. The proposed definition hopes to support the core aspects found in literature, while clarifying the relationship with Mechanics to avoid misinterpretation and filling the gaps surrounding the lack of support to the complex nature of Dynamics.

1. **Its run time behaviour**

   The definition clearly supports the runtime aspect of the concept, an aspect that shows the analytical nature of Dynamics. Since the proposed ontology here is based on a designer perspective, this taxonomy should allow it to be used in development despite its runtime aspect. This is achieved by supporting the domain's characteristic of the iterative design process: it is to be considered as predictable the Dynamics that can be found in any iteration step of the process - i.e. after prototyping and play-testing.

2. **The close relationship between Mechanics and Dynamics, and the direction of influence from one to another**

As the previous definitions here presented also agree, Dynamics emerges from Mechanics, acknowledging the intrinsic relationship among them, and also an order of influence. This idea is contained within the defined taxonomy, and covers one aspect of the proposed ontology: by acknowledging that Dynamics emerge from Mechanics, it embraces the second order of design aspect of the domain - or how the designers control only the Mechanics directly.

3. **The unpredictable nature of dynamics**

   Due to its number of possibilities, it can be harsh to find all Dynamics while developing a game, and as a consequence causing this concept to be not very useful within a designer perspective - which is the objective of this study. The proposed taxonomy hopes to avoid this issue by identifying as Dynamics only the predictable behaviours, giving more emphasis on the designers point of view of the taxonomy. Since the concepts here are iterative design based, we consider as predictable Dynamics the ones that are found either before or after prototyping, as the complexity of the games nowadays can make it almost impossible to predict all dynamics that one user can identify within the game world before set it in movement.

4. **Complexity**

   As it was shown, the dynamics can emerge from different forms within a game experience. There are virtually unlimited possibilities afforded to the player backed up by the exponentially growing complexity of games nowadays, making the finding of all dynamics unachievable. This complex nature of dynamics can sometimes impair the team to work towards describing and refining them during the design process. With an objective of clarifying the possible dynamics and how to work with this taxonomy, this thesis proposes two categories of dynamics: Simple and Complex.

   - **Simple Dynamic**

     Simple Dynamics are the ones emerging directly from one or more Mechanics only, and usually are defined in the early stages of design. Most of this Dynamics can be considered implied within the current game domain - they are contained in the genre of the game, and usually emerged from core or implied Mechanics. In a regular FPS game for example, a player has a mechanic to shoot, equip weapons and reload. The understanding of a killing enemies Dynamic can be considered implied within the game domain, and since it emerges from Mechanics only, is to be considered Simple. It has to be pointed out that not all Simple Dynamics are implied by the development team, as it can require a previous knowledge of the game domain to define it. Using the same FPS game as an example, the player can also have the Mechanic of crouch. Added to the others, it can emerge Dynamic of staying crouched in a position waiting to surprise an unaware enemy, i.e. "camping" (see Fig. 7.4), and this Dynamic can be implied by designers that are familiar with the genre, but for some others maybe not. Nevertheless, since "camping" derives from Mechanics only (crouch, shoot, equip weapon), it is also considered Simple.

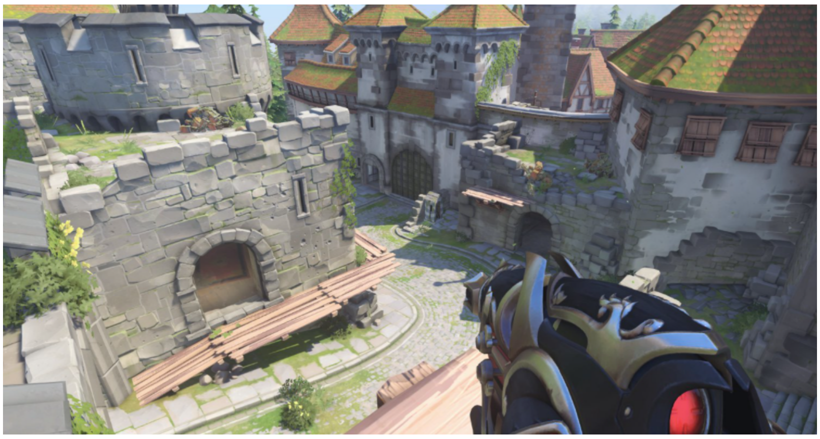


Figure 7.4: Overwatch player camping [37]

These Dynamics are easier to define, since they usually derives from implied or core Mechanics, and can sometimes also be considered implied within a specific game genre. These Dynamics are also easier to refine: since they emerge only from Mechanics, tuning these Mechanics will directly affect this Dynamic.

- **Complex Dynamic**

  Complex Dynamics are the ones that involves other Dynamics in its creation, and as such its complexity can be exponential, and virtually unlimited. Although it may appear that this kind of Dynamic would be mostly unexpected in the first steps of the design and requires prototyping or playtesting to define them, this is not always true. There are plenty of expected

Dynamics in the early stages of development that fall into this category. One example are quests in an RPG game - it can involve multiple Simple or Complex Dynamics, such as killing monsters, levelling up, hunting, gathering loot, equipping armours, among others - and its certainly is defined in the early stages of design.

Another example to illustrate the concept of Complex Dynamic is to imagine a hypothetical multiplayer FPS shooter, where the player has the regular Mechanics of shooting, crouching and equipping weapons, and also a Mechanic to re-spawn at a certain spot after getting killed. The player can crouch and cover while standing still to hide and shoot enemies, resulting in a “camping” Simple Dynamic, which in this hypothetical example, was an early stage expected Dynamic. From the sum of this camping Dynamic with the re-spawn Mechanic, an unwanted behaviour emerged: the player could "camping" in a specific area close to the re-spawn to instantly kill enemies when they appear, without giving them time to fight back. This Dynamic can certainly impair the wanted Aesthetics of the game, in this case Challenge - and by doing so the team should work on this.

7.3 In Use

With the proposed taxonomy, we hope to give the designers a guide to work with the Dynamics in the development process. The first step is to work with the expected simple Dynamics, usually defined at the early stages of the development process.

Using the hypothetical single-player FPS game as example, we have the simple Dynamic of killing players (emerged from shooting, equipping weapons) defined in the very first step of the development process. Since the team knows which are the expected Aesthetics in this game, they can work on refining this Dynamic towards the defined emotional objective. If it’s a game that works on invoking Fantasy, the team can refine the Mechanics associated in the creation of this Dynamic to support this objective: they can choose not to show "floating UI" (i.e. Fig. 7.5) that represent damage or life points merged within the world game when an enemy gets hit, since it can break the immersion of the player with the game world and impairs the "make-believe feeling" necessary to invoke Fantasy. If the same game has the purpose on invoking Challenge, the team could create a complex Dynamic of combos that emerges from killing enemies:

the player would win extra rewards by killing many enemies in a row (i.e. Fig. 7.6). If the game Aesthetics aim towards Sensation, the team could improve the Mechanics that emerges it, like improving the graphical effects of enemies dying – by creating more details or many different animations (i.e. Fig. 7.7).

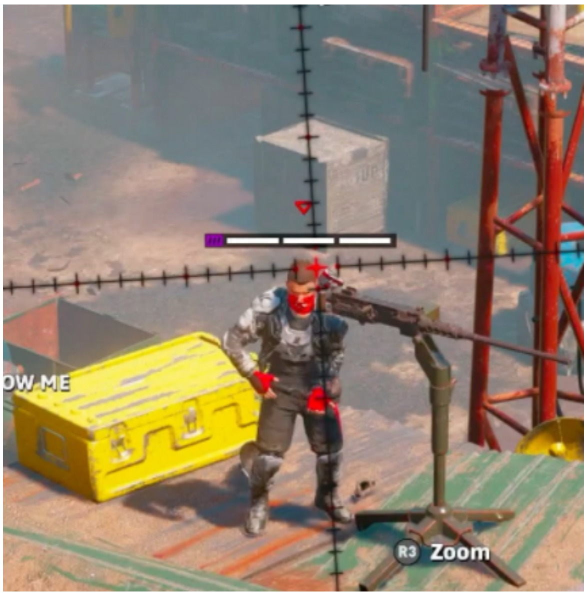


Figure 7.5: Far Cry New Dawn – combat scene with floating UI [31]

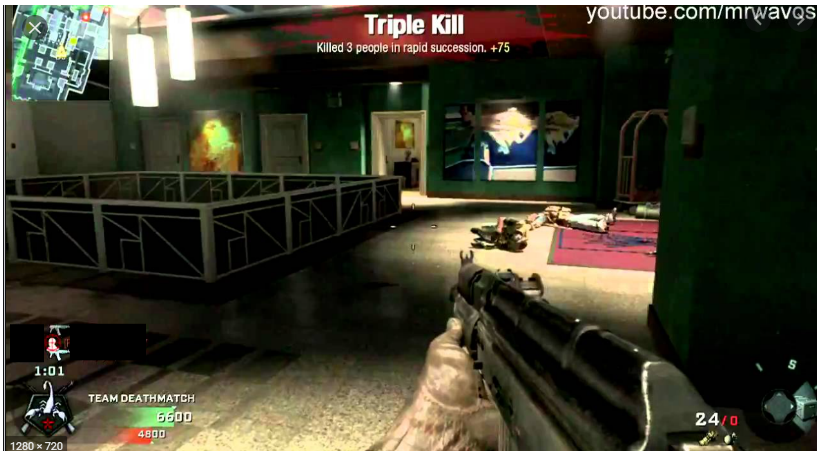


Figure 7.6: Call of Duty Black Ops – triple kill feedback [42]

After working on the expected Simple Dynamics, they could move to expected Complex Dynamics, if there are any in the game being developed – specially if these Dynamics emerges from the already defined Simple ones. The way of dealing with them is similar to the first step of the process, although it is a more complex process

Figure 7.7: Doom Eternal - detailed graphics to enhance Sensation Aesthetic [30]

since there can be two or more layers between the designer and this complex Dynamic: Mechanics and one or more other Dynamics. Using the previous FPS single player game as an example, let's suppose that the player has the Mechanic of upgrade his weapons, and with this new Mechanic a Complex Dynamic is expected in the early stages of development: to hunt for supplies needed to upgrading his weapons (emerged from killing enemies Dynamics and upgrade weapon Mechanic). If this game has Expression as a prioritized Aesthetic, the team could for example enhance the variety of upgrades available, with different colours and combinations, in a way that the player could express himself by it. This is achieved by refining the upgrade weapon Mechanic, but it can also be enhanced by refining the killing enemies Dynamic involved in this complex Dynamic: the team could work on tuning enemies in order to make them not to be a difficult challenge, allowing the user to upgrade his weapons with the solo purpose of expressing themselves, as defeating enemies would not be highly affected by it.

Let's make this last FPS example game a multiplayer cooperative game, and one of the Aesthetic priority is Submission, while Challenge is not an important one. The team noticed that players who are not familiar with the genre do not play for long, since they are not good enough to aim and shoot enemies as their experienced peers - too much Challenge involved. Developers could work with the killing enemy Dynamic to avoid this issue: by creating new shooting Mechanics, the killing enemies Dynamics can be expanded to support different players. The team could for example create a weapon that does not require too much aim, e.g. grenade launcher, that explodes in a large radius and hits enemies, or even melee weapons such as swords or axes that would also not require previous FPS experience to hit enemies. Or they could create new skills or perks that help aiming (e.g. guided bullets) or make enemies bigger and easier to shoot. There are several ways that this issue could be solved, and understanding what Dynamics is and how it is invoked will surely be a more efficient way to deal with it.

The next step of the process is dealing with the emergent nature and the unpredictability of the domain: how to deal with the Dynamics emerged in prototyping.

7.3.1 Refining Dynamics

As proposed, Dynamics are only the predictable behaviours that can emerge, either before prototyping or in any iteration of the designing process. What is to be done with the unexpected behaviours that are found? In the proposed taxonomy, we defined that Dynamic has the purpose to evoke Aesthetics. Based on this, when unexpected behaviour is found, it should be analysed regarding the Aesthetics in play, so the team can decide how to deal with it. The emerging Dynamic can be either removed, ignored or maintained based on its influence in the overall expected Aesthetics.

It's important to note that the following examples using real games is a hypothetical way of looking at issues within the proposed ontology, and I do not claim that this was how the designers approach it; neither was this solution endorsed by them.

7.3.1.1 Ignore

The Dynamic should be ignored when it does not influence the proposed Aesthetics nor has potential to improve them, so it would not be necessary to refine it. Due to the fact that there are so many possible emerging Dynamics, expecting to be able to work with all of them is almost naive, therefore ignoring is a practice that is to be used in some cases.

We can find an example in the famous game Super Mario World. In this game a player can find a secret level to get extra life (see Fig. 7.8), and keep repeating this process for as long as he wants, creating a Dynamic of re-entering this level to easily obtain infinite lives. At first glance one may think that this Dynamic surely needs to be removed, as it directly impacts the Challenge Aesthetic of the game - by allowing the player to never reach zero lives and the "game over", i.e. losing the game. Although it could somewhat impair the Challenge Aesthetic, Fantasy and Explore are probably a bigger priority in this game, and as such the infinite life Dynamic would not directly oppose any of these. In fact, the Dynamic could even enhance these Aesthetics: it provides a chance for players to reach new levels that maybe they would not be able to do it without these extra lives, directly enhancing the wanted Aesthetics.

Figure 7.8: Secret level in Super Mario World [33]

Another example is a game developed by Bethesda Studios called Elders Scrolls Skyrim. In this game the player controls an avatar in an extremely well detailed 3D world, full of creatures to be defeated, caverns and secrets to be discovered and quests to be fulfilled. Exploring and Fantasy are certainly priorities in this game. Regardless of how many Dynamics the designers expected and created for the player, we must remember how the player is the ultimate creator of its own experience: the game is nothing but a tool. The player can create a Dynamic such as trying to reach the highest point of the map, only for the sake of doing it (see Fig. 7.9). This is (probably) not an expected Dynamic, and it can surely be ignored since it neither has potential for being refined to support the wanted Aesthetics nor impairs the player to achieve them.

The first example describes how an unexpected Dynamic can be ignored since it does not impair any Aesthetics objectives of the game, and may even increase its balance. This Aesthetic balance will be detailed in the next chapter. The latter example is an extreme case of unexpected emerging Dynamic: the number of possible behaviours such as this one is virtually infinite. By working on finding these kinds of dynamics can cost an unnecessary amount of time, since this is the category where the unlimited possibilities of Dynamics fall. These types of Dynamics usually emerge during play-testing, and the team should analyse them carefully to avoid spending time on refining

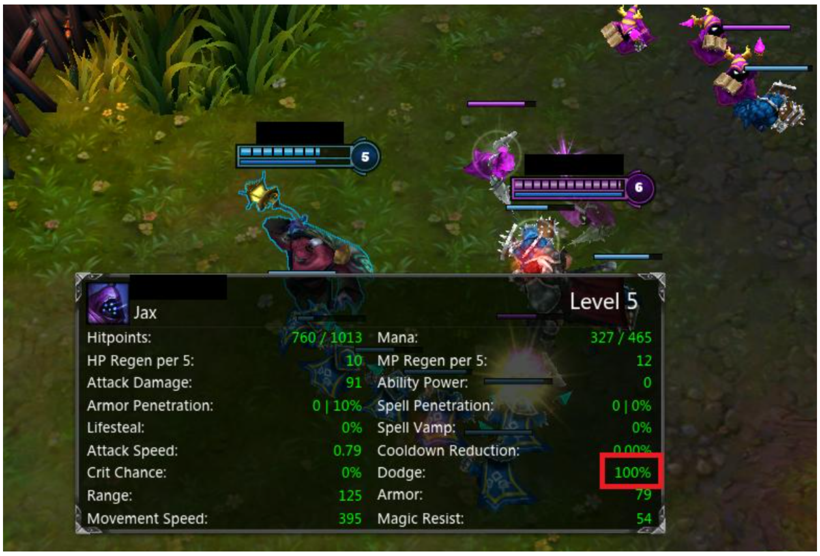
Figure 7.9: Player reaching the highest point in Elders Scrolls Skyrim [34]

a Dynamic that will rarely emerge in most case scenarios and has almost zero influence on the wanted Aesthetics for the game: ignoring it is almost always the best approach.

7.3.1.2 Remove

When a Dynamic can impair the achievement of an Aesthetic objective of the game, it should be removed. This can be done by removing the Dynamics and/or the Mechanics that invoke it.

In a game called League of Legends, by Riot games, the player can choose among several heroes to play as a team against other players online. These heroes have stats, like attack damage and defence. They also had a percentage of chance to dodge an incoming attack, which can be enhanced by the obtention of items that increase this chance. One of the heroes that the player could select, had a particular skill that increased his dodging chance, and it would add to the chance obtained by items. This created a Dynamic where player could select this hero, buy particular items and achieve a 100% dodging chance, therefore making him invulnerable to attacks from other players (see Fig. 7.10). This Dynamic turned out to create an extreme unbalanced advantage to the players who did it, which hugely affected the Challenge Aesthetic of the game - a priority one. In this case, the game workaround for this was to remove this Dynamic from the game by removing the Mechanics that would invoke it: the avatars no longer have the Mechanic of dodging, and the items do not have the Mechanic to improve dodging skills, and as result the unwanted Dynamic was successfully removed from the game.



[redacted]

[redacted]

[redacted] Jax Level 5

Hitpoints:	760 / 1013	Mana:	327 / 465
HP Regen per 5:	10	MP Regen per 5:	12
Attack Damage:	91	Ability Power:	0
Armor Penetration:	0 | 10%	Spell Penetration:	0 | 0%
Lifesteal:	0%	Spell Vamp:	0%
Attack Speed:	0.79	Cooldown Reduction:	0.00%
Crit Chance:	0%	Dodge:	100%
Range:	125	Armor:	79
Movement Speed:	395	Magic Resist:	54

Figure 7.10: League of Legends: avatar Jax with 100% dodge change [32]

In a previous example, we proposed a hypothetical FPS game with the "camping" Dynamic, which added to the respawn Mechanic would create an unwanted Dynamic of camping near the re-spawn of enemies and kill them instantly after they appear, without giving them a chance to fight back. This Dynamic heavily impairs the Challenge Aesthetics, by giving an unfair advantage to the "camping" player. This Dynamic surely needs to be removed. By analysing how this complex Dynamic arises - from the camping Dynamic and re-spawn Mechanic - we have a guide on where to work to remove this unwanted behaviour from the game. Since it's not optimal to remove the dynamic of "camping" without taking out some core aspect of FPS games, neither to remove the re-spawn points of the game since players need to appear somewhere, the designing team has to carefully work in a way to approach this issue. One solution is to change the Mechanic of re-spawning: the players would appear in an area that the enemy team cannot access. With this approach, both the camping Dynamic and the re-spawn Mechanics remains in game, but the unwanted behaviour was successfully removed.

Tibia is a MMORPG where the player controls an avatar in a huge and detailed fantasy world, with a variety of monsters and with the objective of becoming stronger. Challenge and Explore can be considered the Aesthetics priorities here. The game has a Dynamic of hunting where the avatar increases its level by defeating monsters that gives the player experience points. In this game world, when a player kills a monster, he has to wait some predetermined time for the monster to respawn again, and this time ranges from some seconds to several hours. This re-spawn time can make it difficult for the player to keep killing the same monster at the same spot and increase his level, and from this the hunting Dynamic has a characteristic that the player has to keep looking for new places to find another monster's respawn in order to level up, enhancing the Explore Aesthetic. Another characteristic of some monsters in this game is that they can summon other creatures to help them in battle. With the monster's Mechanics to summon, an unexpected Dynamic was found: the player can choose not to kill the summoning monster, and only kill its summoned creatures, for as long as he wants. Since the summoning time is way less than the re-spawning time, the player would increase its level faster than those who do not use this Dynamic, drastically impairing the Challenge Aesthetic by allowing this unfair advantage between players. This Dynamic also goes against the Exploring Aesthetic: the player could hunt always in the same place.

In this game, the Dynamics and Mechanics that invokes this unwanted behaviour belongs to the core of the game genre, and as such it is not possible to remove one or more in order solve this issue. The designers should not remove the summoning nor the re-spawn time-delay Mechanic from the monster, and less so to remove the Dynamic of killing monsters or hunting from the player. This issue is a complex case, since it is a massive multiplayer game with a strong Challenge Aesthetics: it had to be solved in order to avoid misuse and creation of an unfair advantage for some player using it. These complexes cases requires extra caution: even small changes in Mechanics can cascade through Dynamics [1] [16] [6] and changes in Dynamics can change and/or create new unexpected behaviours inside the game - an exponential complexity. The designing team could have come up with a solution like limiting the number of summoning monsters that could be summoned. That could work for the unwanted behaviour of hunting the same summoned monster over and over again, but could also elicit other problematic behaviour. For instance, one player with a higher level could keep killing the summons until the monster reaches its limit, then let a lower level player - who should not be capable of killing the monster - kill it, since there are no summons to support the monster. And this new Dynamic would also impair the Challenge Aesthetic.

In this example we can see how a detailed analysis around refining Dynamics is important. Having to predict possible Dynamics is not an easy task - but it can be simplified if the team understands the basic concept of how it emerges and how it influences the final Aesthetics. The solution created by the designers was to keep all Mechanics and Dynamics that emerged the behaviour the same, but to remove a characteristic of the summoned monster entity: it would no longer afford experience points to the player. This change completely removed the unwanted Dynamic of levelling up only using summons, since its no longer possible – while keeping the Mechanics and Dynamics that are considered important to the game.

7.3.1.3 Maintain

When an unexpected Dynamic arises that has potential to support wanted Aesthetics, it should be maintained, and refined if necessary. Some behaviours that emerge during the development phase are unexpected ways of improving the quality of the game towards the wanted Aesthetics, and as such they should be maintained. But there is also emerging Dynamics that goes against the Aesthetics, but they belong to the core of the game and have to be maintained. In these cases, refining this Dynamic is necessary to stir this behaviour towards the wanted objectives.

In the game League of Legend, the map consists of three lanes where the team can move to defeat the enemy team. During some online matches emerged a Dynamic where the players would agree to only play in the middle lane, and by doing so intensifying and shortening the match, since all the players would fight together all times in the same spot. This Dynamic proved to be an enhancement for the wanted Challenge and Fellowship Aesthetic for the game, and the developers embraced it: they created a new map (see Fig. 7.11) which there is only one lane, and the game is more intense and shorter. This is an example of how an emerging Dynamic can be maintained if it supports the game wanted Aesthetics.

If we us as an example a classic MMORPG game, the Dynamic of hunting for loot or levelling up is usually defined at the beginning of the development process. If the game has Exploring and Challenge as Aesthetics priorities, the team should work on these Dynamics toward the wanted emotional responses: they could create different hunt areas and monsters as an incentive for the player to explore uncharted territory and balance the enemies and their rewards to a fair Challenge between players that hunt in different places. In games like this is common to emerge a Dynamic where the player would only hunt a specific monster with a specific weapon or skills due to easier rewards when comparing it to other hunting places, and as such directly impairing the

Figure 7.11: League of Legends new map [36]

Explore Aesthetic, since the players would not have the incentive to find new hunting areas. This Dynamic would also go against the Challenge Aesthetic: players who are not aware of this behaviour will be in an unfair disadvantage.

In this scenario, it is not optimal to remove the Dynamic of hunting that specific monster using a specific weapon: the designing team worked on creating and modelling them, and reducing the variety of monsters and weapons in a game can reduce the Explore Aesthetic. A solution that is usually adopted is to change some entity characteristics (or knowing responsibilities) involved, but not the Mechanics afforded to them, and as consequences, the Dynamics that emerged from them will be kept. In this case, the designers could change the health of the monsters to increase difficulty or lower the experience points afforded to players that kill it - in both cases, the Dynamic of hunting that monster will be kept - but the unfair advantage is no longer within it.

7.4 Summary

The proposed definition was based on how Dynamics was defined by Le Blanc et.al [1], and it embraces the previously described characteristics of the game design domain, while maintaining the aspects of this taxonomy found in literature.

Dynamics is the bridge that connect the designers to players. It's what emerges from designers actions and creates the emotional response on players. Having a clear understanding of how Mechanics are the foundation for Dynamics, and how Dynamics work on creating Aesthetic, the development team can have a clearer path to follow in order to achieve the game's emotional purposes. Correctly defined Dynamics will support the development team by showing them where and how to work, increasing efficiency in the designing process.

.

Chapter 8

Aesthetics

"Aesthetic arises not from the game alone, but the combination of game and player. [25]"

The last letter in the MDA framework stands for Aesthetics. The following sessions will analyse this proposed taxonomy by Le Blanc et. al. [1] and show how this definition is received by the domain's literature. This chapter will dive into the issues that surround this concept: the widespread misconception of what it is and the difficulty task of defining it due to its subjectiveness, and it will finish by proposing a new definition, supported by examples of how it can improve the game designing process.

8.1 State of Art

When it comes to the Aesthetic concept of MDA framework, the first thing that arrives is a misconception of the term, due to its various meanings in different fields of studies. For instance, on phenomenology, Aesthetics ".. is trying to answer the question of HOW we perceive things.", and is also a term used in psychology: "[...] having to do with how different people perceive the same colour, sound, melody, picture or text in completely different ways, as well as trying to understand the reasons and implications behind those differences" [6]. When it comes to the game domain, the misconception is even clearer: it has a widespread association with the visual style, graphics, or the art style of a game, as acknowledged by Frank Lantz: "No matter how strictly you attempt to clarify that you aren't talking about visual Aesthetics, that you are talking about the broader set of qualities that make an experience beautiful, meaningful, interesting, valuable, etc., people will continue to naturally, instinctively think of “Aesthetics” as “visual Aesthetics” [24].

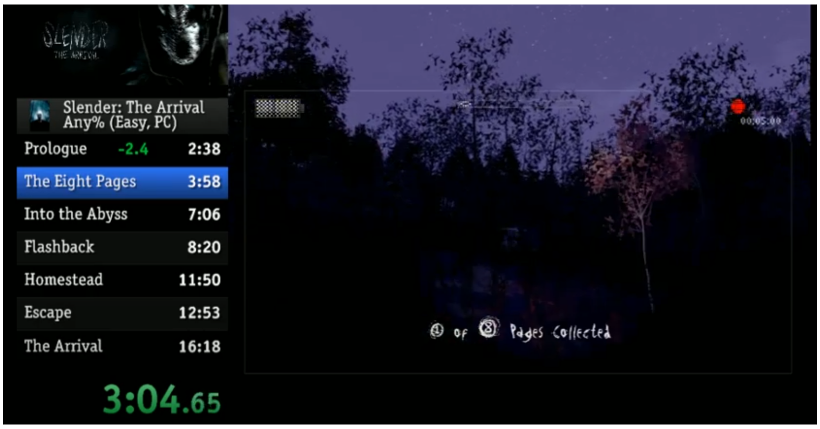
Trying to explain the “fun” of playing games is not an easy task - and as the MDA authors stated themselves, this is not the objective of defining Aesthetics: “the point is just to have something you might use as a vocabulary of play Aesthetics. Once you have the vocabulary you can get past 'fun' and start doing an analysis of fun. The particular feeling a game conveys" [1]. The taxonomy that MDA presented us is:

“Aesthetics describe the desirable emotional responses evoked in the player, when they interact with the game system.”

The first aspect to notice in this definition is how there is no direct relation with the visual or art style of a game - but with the emotions that the game can invoke. The second point around this definition is how it contains the *runtime* characteristic of the game domain merged with it: "when they interact with the game system". By embracing the *runtime* aspect, Aesthetic deals with the experience emerged when the player interacts with the game, and encourages an experience-driven (as opposed to feature-driven) design [1].

The issue that surrounds this definition is that the authors seem to imply that the game has an objective to directly evoke emotions in the user - “Evoked in the player”. As mentioned before, the game is a ludic device, which allows the players to create their own emotions, and not directly invoke them. The designers will aim for expected emotional objectives, but the truth is that the player has the ultimate decision of which emotion they will try to achieve by interacting with the game.

This becomes clear when we talk about “speed runs”. Lately in the game community, there is a common practice to stream online your live gameplay, and there are people who made a job out of this. A new way of playing has become popular among this community: the *speed runs*. Basically is a gameplay mode where the player tries to finish a game in the fastest time possible (Fig. 8.1). In this mode of playing, the user creates his own Aesthetics: Challenge. Usually it does not matter the content of the game nor the Aesthetics objectives aimed by the development team: the user will ignore it in order to break their time records. The player could *speed run* a game that has Sensation Aesthetics as priority, powered by well composed songs and sound effects, while playing it without sound - so he can concentrate more and finish the game in less time. Or he could play a game with an extreme full detailed 3D world to explore and not chart 10% of it. This mode of playing is an example of how the player is the real creator of his own emotions, and sometimes it happens to be completely different from what the designers had in mind. Nevertheless, these games should not be considered a failure in design at all because some players decide to play it differently. The game is a tool, a ludic device that allows a player to create his own personal experience - and the designers have only partial control over it.



Slender: The Arrival
Any% (Easy, PC)

Prologue	-2.4	2:38
The Eight Pages		3:58
Into the Abyss		7:06
Flashback		8:20
Homestead		11:50
Escape		12:53
The Arrival		16:18

3:04.65

Figure 8.1: User playing the terror game Slender: The Arrival as "speed run"[35]

There is not a clear way of determining what makes a game fun. First, even defining fun is a hard task: one can dive deep into philosophy and psychology and still fail to find a clear definition for it. As games have the implied objective of being fun, developers need a way of better understanding the concept of fun and how to work in achieving it, and this is where Aesthetics can help. As MDA authors mentioned, defining fun is not the point here, but to create a vocabulary that can be used as a compass to lead the team towards the expected player's emotional responses [1] [28]. The way that the authors overcame the harsh task to rationalize fun was to introduce some taxonomies to exemplify it within the concept of Aesthetics, based on Le Blanc's website, "The 8 kinds of fun" [20]:

1. **Sensation**: Game as sense-pleasure: Games that has a strong characteristic of engaging senses - either by visual art style or sound design

2. **Fantasy**: Game as make-believe: Games that create a make-believe world, an alternative reality to the player

3. **Narrative**: Game as drama: Games that has a well written narrative, with defined characters and/or world

4. **Challenge**: Game as obstacle course: Games that have a competitive feeling, that invoke the thrill of competition. It should be noticed that it can also happen in single player games, the the fun arises upon overcoming a difficult challenge
5. **Fellowship**: Game as social framework: Games that one of its aspects is to engage the player into social relations, with friends, family or other gamers.
6. **Discovery**: Game as uncharted territory: Games that motivates the player to explore and discover new features contained in it
7. **Expression**: Game as self-discovery: Games that enable the player to find ways of expressing himself
8. **Submission**: Game as pastime: Games that focus on create a distraction for the player

It's important to note, as mentioned before, that Le Blanc et al. was not trying to define every emotive aspect that can be evoked on the player: "This includes but is not limited to the taxonomy listed here" [1] - but was trying to create a starting point in objectifying emotions involved in game design domain. These taxonomies should be seen as a guide to set the course of the designing aiming for one or more chosen Aesthetics.

8.2 The Proposal

Since there is a lack of literature defining Aesthetics within the game domain due to its subjective nature, we propose only a small change in order to make it clear that the player is the ultimate responsible for invoking emotions. Game is a ludic device, a tool. It does not invoke emotions - it allows the player to invoke it.

"Aesthetics describe the desirable emotional responses that the player can invoke, when interacting with the game system."

8.3 In Use

This taxonomy is a guide to help designers by explicitly defining the emotional purpose of the game, and as such help them when making decisions towards the expected emotional responses. This is the first concept of MDA that should be defined, and the designer has to be aware of how to use it. A game should not pursue all of them - neither aim for only one - but to set some of them as priority and use them as a compass to navigate the designing process towards a direction - the emotional purposes.

Its not enough to say that the team should design a single player first person shooter: without understanding what are the emotional objectives of the game, how would the team prioritize tasks? They should spend more resources on realistic animations or on a detailed narrative? The gameplay learning curve should be steep by adding a variety of weapons and skills? Designers should focus on creating a huge detailed open world? In the real life world, designers have deadlines and limited resources when designing a game. Maybe every designer would want to pursue an excellent and responsive UI, a variety of visual assets and realistic graphical effects, well composed songs, as well a very complete story for their games that is to take place in a huge fantasy world full of characters and secrets to be discovered. And all of this in a massive multiplayer online environment. The cost of creating this would probably surpass the ones defined by the stakeholders for most companies, therefore choices have to be made. And this is where Aesthetics will guide designers throughout these decisions: **Resource Management.**

When the team defines which are the wanted emotional responses in the player, decisions about resources management can be done with an explicitly reason. In the previous FPS game, if the wanted Aesthetics was Challenge, the team should work on a steep gameplay learning curve, spend more time on the creation of a good artificial intelligence for enemies and carefully deal with level design. If the objective was Fantasy, the team could prioritize the visual aspect of the game, trying to make it look more realistic, in order to enhance immersion and create a greater chance of a make-believe felling in the player. When Aesthetics are used to decision making about resources management, there is a greater chance the game will achieve its emotional purposes, and with less cost: improving the design efficiency.

One important aspect of the game domain that has to be understood by designers is that players are not equal. There are different players - and they play for different reasons. A user who enjoys being challenged on hardcore FPS online multiplayer games may not enjoy as much play in order to explore uncharted territories, neither a casual player that enjoys some mobile puzzles as a pastime would enjoy an extremely hard boss that would be a challenge to be defeated. Defining and understanding Aesthetics can deal with this aspect of games: there are **Different Types of Players.** By understanding what is the target of the game, developers will have a better knowledge of why they play games - and based on this design a game that can better fulfil the players expectation.

There are many cases in the industry where game sequels failed to delivered what the players were expecting. Users who were familiar with the prequel played for specifics emotional responses, and without understanding this fact, designing a sequel could fail in allowing the player to invoke these expected emotions, or Aesthetics. If a company is hired to develop a sequel of a game, without the knowledge of how Aesthetics works, they could produce a well designed game and still fail to deliver what the player base was expecting. This does not mean that the sequel should be extremely similar to its prequel: by correctly defining which Aesthetics the game's base player pursue, the team would understand which Dynamics they could enhance or even create in order to make a new game, a different game, that still has the "feeling" of its prequel. Aesthetics works as a way of clarifying what players were expecting. Understanding this will surely increase the acceptance of the prequel player's base.

One example of this is a game called Diablo III was launched in 2008 by Blizzard, as a sequel of its famous Diablo series. Diablo is a game where the player controls an avatar in an alternative terror world, where he has to kill monsters and find better equipment to get stronger, so that he is able to defeat the ultimate boss: Diablo. There was a Dynamic in Diablo II where the player had to hunt multiple times some hard bosses in order to "farm" (i.e. find) rare weapons and equipment, so he would get stronger and finish the game. This "farming hunt" Dynamic invoked Challenge Aesthetics and was at the core of Diablo II - players loved it. When Diablo III was released, Blizzard introduced a new system in the game, an auction house, where players could buy and sell equipment from other players using real money. This created a Dynamic where some players would farm items for real life profit, and impaired the Dynamics of farming hunt from another players: they could easily buy rare items in the auction house, and as such there was no real incentive to "farming hunt", and as consequence created a huge impact on the Challenge Aesthetic. Due to complaints from the community, Blizzard removed the auction house from the game, reinserting the "farming hunt" Dynamic and it is a success until this day.

Another case of failure is Metal Gear Survive, launched in 2018 by Konami. The Metal Gear series is all about stealth and spying: the player controls an avatar that has to silently uncover secrets of the current map to proceed the game. There are numerous stealth Dynamics contained within the series - but not in Metal Gear Survive. This title contained more combat and survival Dynamics, which was not well accepted by the players. This title was not about spying: the player has to build a base to defend itself from attackers, and stealth is barely necessary. The change in the Aesthetics invoked in players resulted in an extremely bad reception by the fan base, and its consider the worst game of the series.

The Aesthetics play a big role in the development process by defining an objective direction to where the team should aim when making decisions. The next session will explain how to define them.

8.3.1 How to Define

The first step when developing a game is to define the core Aesthetic of it - or what is the main experience the game would allow the player to invoke. Sometimes this is a subjective decision: when the game idea arises from an inspiration or dream of a small indie development team for example. The main Aesthetics here comes from the designers, and its definition may be supported by analysing similar games that were sources of inspiration. There are cases where the Aesthetics is pre-determined by the stakeholders. If a team is hired to create a sequel, an *advergame* or an educational *serious game* for example, the Aesthetics can be fixed by the contractors. The idea here is to define one main Aesthetic to support the development, and follow to the next step: defining the secondary Aesthetics.

After the main Aesthetic is fixed, the team should work on defining one or more secondary Aesthetics, as a weight to be counted when decisions are to be made. This step should use some information regarding what is needed to define them, which are, but not limited to:

- **The Knowledge**: The development team has to be completely aware of its capability. With an experienced graphic designers team it would be wise to select Sensation as secondary Aesthetics. A small indie team who are mostly software programmers should not select Narrative as priority, since (probably) they would not have the necessary skills to efficiently achieve it.

- **The Target**: Understanding what type of player the game focuses on is crucial for its success, and its extremely important when defining Aesthetic. By identifying the player's type, the team can analyse already existing games that share the same target and use them as inspiration when defining Aesthetics. A good knowledge of the target is useful in many ways, such as understanding what is the main platform they use for games: it can be counterproductive to focus on Sensation Aesthetic and create extremely detailed graphics if the target mainly plays on mobile phones, as they probably do not have the graphical capacity to render it.

- **The Market**: Using business related information, the team could focus on which Aesthetics are usually expected by its target. What are the mainstream game's Aesthetics? Are there any new and common Dynamic being introduced in games? If so, which are the Aesthetics they hope to evoke? Do players nowadays care more about graphics (Sensation) or story(Narrative)?

- **The Cost**: By having a clear understanding of its capacity, the team should be aware of the costs to achieve each Aesthetics. Which Aesthetic would be easier/less costly to be worked by the team? Which Aesthetics will not be worth it to spend time working?

When first defining the prioritized Aesthetics, the designers know where to put a bigger effort in the designing process. It would not be optimal (for most game companies) to hire an entire orchestra to compose and record the soundtrack of a puzzle game that is to be played as a pastime. Submission is the aimed Aesthetic here, not Sensation. Neither to allocate all the graphic designing team to create an extremely well detailed environment world in an online competitive racing game where Challenge is the priority over Fantasy and Discovery. In other common scenarios, the team should decide to work more in a detailed 3D world over refining the mathematical progression of the player's attacking stats in a single player RPG that main Aesthetics could be Fantasy or Explore, and not Challenge.

When we have the proposed Aesthetics in mind, even small decisions are objectively supported in the development process. In a RPG game for example, the font size and style of the text that shows how many experience points the player won by the Dynamic of killing monsters should be well thought of if Challenge Aesthetics is a priority. If Sensation and Narrative are priorities, the sound designing team could spend more time in recording and editing the voice overs of the NPC's of the game, and directly work on enhancing Dialogues Dynamics by refining the Mechanics that emerges it:

they could improve the speaking Mechanic of a NPC entity by creating more facial expressions and better animations. In a game where the user can buy outfits for his avatar with points that he won by solving puzzles, and Expression is a priority Aesthetic, the designers should first work on a variety of detailed outfits before working on the challenge progression of the puzzles, allowing the players to better express themselves with different outfits.

8.3.2 How to Balance

As it was explained, the designers should aim for more than one Aesthetic priority. These Aesthetics should be well balanced, in a way that they work enhancing each other, and not the opposite. This task of balancing them deeply falls into subjectiveness - there isn't one rule or structured way on how to achieve it. The team has to do a lot of prototypes and play-testing in order to figure out how the Aesthetics would work together for the majority of players, but in the end, the decision has to be made in an analytical environment of virtually unlimited possible dynamics that can evoke human emotions - hence the subjectiveness implied on this task.

A great example of unbalanced Aesthetics was given by Philip Tan [23] analysing a game called Bioshock Infinity, by Irrational Games. In this game, the player is sent to a fantastic and beautiful world, with well detailed NPC's living in it, with a great story behind, supported by small details like posters, NPC's dialogue, well designed architecture, implying Fantasy Aesthetics. Another characteristic of this game is that the player is constantly engaged in combat (Fig. 8.2), being attacked and having to defend himself, which is part of the Challenge Aesthetic. The player has the Mechanics of gathering food on the ground to heal and to pick money and loot from corpses to improve his equipment, enhancing his combat victory chance. This resulted in a Dynamic where the player would keep looking for more food and items laying on the ground or in corpses, and ignoring the beautiful modelled city and all the small details that tell the story of this fantastic world - and were carefully designed. Since the game team worked hard on bringing this Fantasy Aesthetics as a priority over Challenge, this is not an optimal result for the player experience. A supposed way of dealing with this unbalanced Aesthetics is to create map areas without combat or items to be gathered, to encourage the player to focus on the beautifully designed world and its small details. This possible solution would not change the Aesthetics priorities, but balance them in a way that the player would be able to invoke all the expected emotions.

Figure 8.2: Bioshock Infinity: combat ahead [39]

In a previous example, Super Mario World allowed an infinite life Dynamic that could somewhat impair Challenge Aesthetics, but it could also support other Aesthetics in play, such as Fantasy or Exploring, by easing the path for the player to unlock more of the game world and the story contained within without losing all his lives. The decision here is subjective, since there was not a structured way of identifying how much this change in Challenge Aesthetic would affect others - it relies on prototyping and play-testing, and ultimately on the designers perspective of the issue.

Imagine a FPS game that Challenge and Narrative are the prioritized Aesthetics. Users would play a campaign with a detailed story that is presented as cut-scenes in between levels, while trying to defeat hard enemies in a frenetic fast-paced shooting combat. We can predict some ways that these Aesthetics can work against each other: if the designers create long non-skippable cut-scenes, the players that are hooked by Challenge invoked by the fast-paced combat Dynamic could get bored - and if we allow them to skip the cut-scene, the narrative would not be well presented. Understanding this aspect of balancing Aesthetics can help designers to avoid these issues. In this case, they could do small change - such as create short cut-scenes, or new Dynamics to present the story, such as a more complex dialogue system where a NPC would talk to the player during combat, so it would not bore the player nor the player would not receive the narrative.

Destiny 2 is a multiplayer massive first person shooter, created by Bungie. In this game, players can invite their friends to form squads and join forces to defeat enemies, indicating a Fellowship Aesthetics. When defeating enemies, the players gain experience points, find new weapons and items that can be used to increase the power of his equipment, so that he can defeat stronger enemies. This levelling up Dynamic suggests the Challenge Aesthetics. In many similar games, there is an implicit problem that can impair the fellowship Aesthetics: players can become way more powerful than other players - so it would not be possible for them to join forces in a fair way, and the Fellowship is somewhat affected. Destiny 2 has a way of dealing with this: the power of your weapons do not increase much when you level up. The advantages of levelling up is that you can find different weapons, items and skills that fit your gameplay style, and/or changes some visual attributes of your items - but they do not become much stronger than other items and weapons. This change in the levelling up Dynamic created a more balanced gameplay between players that are lower level than others, and as consequence improved the balance between Fellowship and Challenge Aesthetics.

Game designers must understand that there is a difference between balancing Aesthetics and changing the public target of a sequel. The previous example of Metal Gear Survive was not well received because the public's expectations was the same as the previous tittles of the series: stealth and spying. There are times when a sequel explicitly changes the Aesthetics due to a new objective, and it can become a success. In the Metal Gear series, there is another game that do not follow the stealth/spy gameplay, called Metal Gear Rising: Revengeance, launched in 2013. This game is all about hack'n'slash: the avatar has a sword and has to kill multiple enemies performing combos in a frenetic gameplay - far from the stealthy gameplay from the previous Metal Gear games. And differently from Metal Gear Survive, was a success. The players knew what to expect: a different game, with a different emotional responses and a different gameplay, that happens in the same universe as the Metal Gear series.

8.4 Summary

Aesthetic is a term loaded with various definitions - and it can contradict itself in different fields and in different forms. This issue around its definition goes against the purpose of this study - a clear and defined ontology - as agreed by [6] "Any practical game design framework should have at its core the goal of helping game designers understand how to approach their daily work. It should help frame the development process, while also accurately representing the underlying art form. A framework loaded with philosophical terms that are only useful for assessing art after it has been created is by definition of no practical use.".

The idea here is to find a way of defining emotions without diving too deep in human physique - but having some kind of guide when it comes to the proposed emotions that the game should allow the player to evoke. The use of the taxonomies provided by Le Blanc in his website [20] was used as a vocabulary in the MDA paper and here, but it does not mean that is the only correct way of describing Aesthetics. For instance, the Aesthetic of Challenge could be broken down into Multiplayer Challenge and Single Player Challenge, if the knowledge of the development team supports this and if this helps sharpen the decisions that have to be done during the development process. Serious games could have his educational purposes defined as one Aesthetics, and used like the others: as a compass to guide the development.

Understanding Aesthetics and how developers can invoke them by correctly following the proposed pattern (e.g. fig. 8.4) will ease the process of dealing with unexpected end emotional results: it becomes clear which Mechanic should be directly changed to improve the end result. Not only would help dealing with unexpected results, but also increase efficiency when creating new Aesthetics due to the clarity of how one layer would affect the next layer - Mechanics (Core, Implied, Extra) creates Dynamics (Simple, Complex) that invokes Aesthetics.

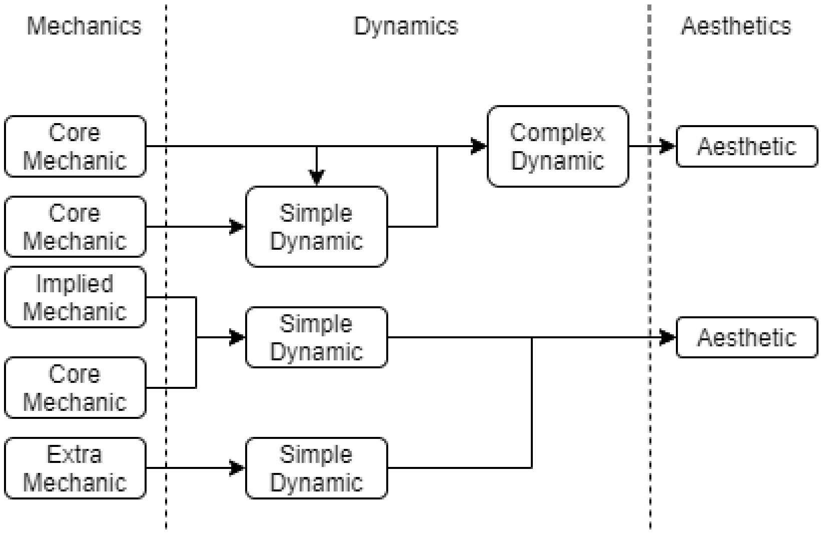


Figure 8.3: Proposed diagram

The use of Aesthetics allows the developers to refine the game experience towards the desirable result, although the step of balancing Aesthetics can fall into the subjective category - as the emergent nature of the game implies that different players have different experiences towards the same game. A detailed knowledge of the public target can lessen this issue, although its complete understanding will remain elusive due to its subjective nature. We must remember that the player is the ultimate creator of his own experience: the game is a tool, a ludic device. It does not matter how many hours were spent trying to balance a game based on Explore and Fantasy; there will always exist that one player who will create its own Challenge Aesthetic by playing it as "speed runs", and this do not lessen at all the importance of defining Aesthetics during the development process.

Chapter 9

The reason why

"Tell me why" - Backstreet Boys

9.1 Conclusion

The aspects that surround games regarding its emergent and ludic nature added to the increasing complexity of nowadays digital games in a relatively new domain harshen the creation of a clear and structured ontology that embrace these aspects. The lack of a unified accepted ontology impacts the ability of the game domain to grow. An ontology can improve the understanding and teaching of the concepts contained within the domain, and allow designers to create methodologies to support the development process and enhance the quality of their products. Despite that many attempts were made, the domain still contains this gap: its basic concepts are not well defined. And this is the first gap that must be closed in order to create a designing methodology.

The difficulty of acceptance of a structured design methodology is a common attitude by designers that belongs to domains where creativity is at the core of the creation process - such as music, movies, literature and games. This fact added to complex aspects of designing games previously described can make it hard to create a design methodology. This thesis tried to show how a structured methodology that is supported by a clear ontology will not impair creativity at all - but maybe even enhance it. By connecting the Aesthetics objectives to all the abstraction layers of the game being developed, I hope to justify decisions of design - as little as camera movements in a terror game, or as big as changing the core Dynamic of a game - to an Aesthetic objective, or to a emotion response the player can invoke. By working as a guide to support designers in their creativity process, they can correctly aim their creation towards the expected end result of the product.

It is a difficult task to achieve a structured methodology in a domain that has so many specific characteristics and aspects that have to be supported by it. It's understandable that game designers would not easily adopt one. I hope that this thesis moves the domain a little closer to this objective and allows designers to enhance the quality of the development process and how fun the game is for their players.

Because this is what a game is. A ludic device to bring joy to the user, a tool to evoke emotion - and that should be clear in every step of the designing process. This is the reason why.

9.2 Future Work

The MDA framework pointed out that the eight kinds of fun are a starting point towards a vocabulary to be used as a guide to understand the player's emotions. By revealing more about this subjective area of the game domain, designers will have a better understanding about the emotional objective of the game, and hopefully increase its quality. A paper written by Roberton Dilon [15] enhanced the way that designers could work on player's emotion. The author created what he calls as The 6-11 framework, a methodology that could be used alongside MDA that focuses on six emotions and eleven instincts that are recurrent in psychology: Fear, Anger, Joy / Happiness, Pride, Sadness and Excitement, and the instincts are Survival (Fight of Flight), Self Identification, Collecting, Greed, Protection / Care / Nurture, Aggressiveness, Revenge, Competition, Communication, Exploration / Curiosity and Colour Appreciation. The extra detailing about players emotions is a reasonable way to further increase the efficacy of game design methodologies, and future work is needed in this area of the domain.

As previously stated, this thesis was born out of necessity. The way that I stumbled across the lack of an unified ontology and design methodologies for the domain was when I was researching serious games development. Although there is literature about this topic, they are not supportive among themselves, and sometimes contradicts one another. The lack of a domain's ontology is one main reason for this. The increase of serious games within the industry is evident, and I believe that this thesis could support their development as well. If successful, I hope that the proposed methodology derived from MDA will allow a better understanding of the serious objectives of the game (e.g. educational or advertising) and as a consequence increase its efficiency.

Bibliografia

[1] Robin Hunicke, Marc LeBlanc, Robert Zubek. *MDA: A Formal Approach to Game Design and Game Research*. Prepared for Proceedings of the Challengs in game AI workhop, Nineteenth National Conference on Artificial Intelligence, San Jose, CA, July 25, 2004.

[2] David Callele, Eric Neufeld, Kevin Schneider. *Requirements Engineering and the Creative Process in the Video Game Industry*. Department of Computer Science University of Saskatchewan Saskatoon, Saskatchewan, Canada S7N 5C9.

[3] Joris Dormans. *Engineering emergence: applied theory for game design*. Interfacultary Research Institutes Faculty of Humanities, January, 2012.

[4] Staffan Björk, Sus Lundgren, Jussi Holopainen. *Game Design Patterns*. Conference: Digital Games Research Conference 2003, 4-6 November 2003, University of Utrecht, The Netherlands.

[5] José P. Zagal, Amy Bruckman. *The Game Ontology Project: Supporting Learning While Contributing Authentically to Game Studies*. College of Computing, Georgia Institute of Technology, 2008.

[6] Wolfgang Walk, Daniel Görlich, Mark Barrett. *Design, Dynamics, Experience (DDE): An Advancement of the MDA framework for Game Design*. March, 2017.

[7] Jesse Schell. *The Art of Game Design - A Book of Lenses*. CRC Press, 1st edition, August 4, 2008

[8] Joshua Krit, Eduardo Mangeli, Geraldo Xexéo. *Building an Ontology of Boardgame Mechanics based on the BoardGameGeek Database and the MDA Framework.* Universidade Federal do Rio de Janeiro

[9] Glauco Ofranti Trindade. *Video Game Development Ontology*. Universidade Federal do Rio de Janeiro, December 2015

[10] José P. Zagal, Michael Mateas, Clara Fernández-Vara, Brian Hochhalter,Nolan Lichti. *Towards an Ontological Language for GameAnalysis*. 2005, Vancouver, British Columbia, Canada

[11] Megan A. Winget, Walker Sampson. *Game Development Documentation and Institutional Collection Development Policy*. 2011, University of Texas at Austin, School of Information

[12] Frank Malcher, André M. M. Neves, Leo Falcão. *Aplicação do Game Ontology Project no Processo de Análise de Similares para Design de Jogos*. 2009, VIII Brazilian Symposium on Games and Digital Entertainment

[13] Richard Rouse III. *Game Design Theory & Practice - Second Edition*. 2005, Wordware Publishing, Inc

[14] Sylvester Arnab, Theodore Lim, Maira B. Carvalho, Francesco Bellotti, Sara de Freitas, Sandy Louchart, Neil Suttie, Riccardo Berta and Alessandro De Gloria. *Mapping Learning and Game Mechanics for Serious Games Analysis*. 2015, Serious Games Institute, Coventry University, Coventry, UK

[15] Roberto Dillon. *The 6-11 framework: a new methodology for game analysis and design*. DigiPen Institute of Technology, Singapore

[16] Miguel Sicart. *Defining Game Mechanics*. 2008, Game Studies (http://gamestudies.org/)

[17] Luiz Claudio Silveira Duarte. *Revisiting the MDA framework*. Accessed March 29, 2020. https://www.gamasutra.com\/blogs\/LuizClaudioSilveiraDuarte/20150203/233487/Revisiting_the_MDA_framework.php

[18] Tabea Iseli (April 17, 2018). *MDA in Game Development (no, not the drug)*. Accessed September 13, 2020. https://ava-game.com/2018/04/17/mda-in-game-development/

[19] Tadhg Kelly(January 18, 2011). *Game Dynamics and Loops [Game Design]*. Accessed October 3, 2020. https://www.whatgamesare.com/2011/01/game-dynamics-and-loops-game-design.html

[20] Marc LeBlanc. *8 Kinds of Fun*. Accessed March 9, 2020. http://algorithmancy.8kindsoffun.com/

[21] Mathew Galant. *Mechanics, Dynamics & Aesthetics*. Accessed November 21, 2020. http://gangles.ca/2009/08/21/mda/

[22] John Martin. *MDA (Mechanics, Dynamics, and Aesthetics) in Canvas*. Accessed November 21, 2020. https://community.canvaslms.com/t5/Instructional-Designer/MDA-Mechanics-Dynamics-and-Aesthetics-in-Canvas/ba-p/275803

[23] *Posted by 11126X*. Accessed November 25, 2020. https://www.youtube.com/watch?v=pD74RdAMW8s

[24] Frank Lantz. (September 14, 2015). *MDA*. Accessed January 18, 2020. http://gamedesignadvance.com/?p=2995

[25] Ian Schreiber. *Game Design Concepts*. Accessed January 18, 2020. https://gamedesignconcepts.wordpress.com/

[26] *The 400 Project*. Noah Falstein, Hal Bardwoord. Accessed September 21, 2020. http://www.finitearts.com/Pages/400page.html

[27] *The Game Onthology Project*. José P. Zagal, Michael Mateas, Clara Fernández-Vara, Brian Hochhalter,Nolan Lichti. Accessed September 21, 2020. https://www.gameontology.com

[28] *GDC: Game Design Workshop: Mechanics, Dynamics, Aesthetics*. Christian Nutt. Accessed September 21, 2020. https://www.gamasutra.com/view/news/108415/GDC_Game_Design_Workshop_Mechanics_Dynamics_Aesthetics.php

[29] *Taken from youtube video Sniper Elite 4 : Brutal Funny X-Ray Moments and Combo Traps !, by PC Gaming Videos, 2016*. https://www.youtube.com/watch?v=h9AOEIfDFZE

[30] *Taken from: NVIDIA and AMD each have drivers ready for DOOM Eternal, posted by Sean Endicott, 2020*. https://www.windowscentral.com/nvidia-and-amd-each-have-drivers-ready-doom-eternal

[31] *Taken from: Far Cry New Dawn review, posted by Steven Petite, February 14, 2019*. https://www.digitaltrends.com/game-reviews/far-cry-new-dawn-review/

[32] *Taken from reddit forum post by IoNJohn*. https://www.reddit.com/r/leagueoflegends/comments/13jlxo/why_is_this_still_here/

[33] *Taken from youtube video How to get to the Top Secret Area in Super Mario World: Super Mario Advance 2, by DaddyGamer Fred. 2016*. https://www.youtube.com/watch?v=qGIr1RbbikU

[34] *Taken from reddit forum post by a deleted user*. https://www.reddit.com/r/skyrim/comments/mrf9v/the_highest_point_on_nirn/

[35] *Taken from youtube video Slender: The Arrival - Speedrun Any% Easy in 16:14 [Old pb], by FlowerSpeedrun, 2016*. https://www.youtube.com/watch?v=nnnAo97JMOw

[36] *Taken from the official website*. https://na.leagueoflegends.com/

[37] *Taken from: The Best Sniping Spots For Widowmaker On Eichenwalde*. https://www.esportstales.com/overwatch/best-widowmaker-sniping-spots-eichenwalde

[38] *Taken from: The Sylverian Dreamer, Blizzard's Plea To Have Faith In 'World Of Warcraft'*. https://www.forbes.com/sites/hnewman/2019/06/27/the-sylverian-dreamer-blizzards-plea-to-have-faith-in-world-of-warcraft/?sh=45fb1b8d7ca9

[39] *Taken from: Video game: Bioshock Infinite, posted by Pavan Shamdasani, 2013*. https://www.scmp.com/lifestyle/technology/article/1207642/video-game-bioshock-infinite

[40] *Taken from youtube video Pokemon Yellow Gameplay - Ep 13, posted by pikachulover306, 2008*. https://www.youtube.com/watch?v=PHZcYAiw5vs

[41] *Taken from CyberPunk gameplay, version 106, December 2020.*

[42] *Taken from youtube video Escalation Map Hotel, Triple Elevator Kill, Call of Duty Black Ops (Wavos), posted by MrWavos, 2011.* `https://www.youtube.com/watch?v=YXdi-0dUo0ks`

[43] Michael Gruninger, Olivier Bodenreider, Frank Olken, Leo Obrst and Peter Yim *Taken from Ontology, taxonomy, folksonomy: Understanding the distinctions.* 2007, Ontology Summit

.