U.PORTO

MESTRADO
MULTIMÉDIA — ESPECIALIZAÇÃO EM TECNOLOGIAS INTERATIVAS E JOGOS DIGITAIS

Visual Aesthetics in Digital Games: A Comparative Analysis Between Photorealism and Stylized Graphics

Pedro Lopes Negrão de Lima

M
2022

FACULDADES PARTICIPANTES:

FACULDADE DE ENGENHARIA
FACULDADE DE BELAS ARTES
FACULDADE DE CIÊNCIAS
FACULDADE DE ECONOMIA
FACULDADE DE LETRAS

U.PORTO

Visual Aesthetics in Digital Games: A Comparative Analysis Between Photorealism and Stylized Graphics

Pedro Lopes Negrão de Lima

Master in Multimedia, University of Porto

Supervisor: José Miguel Santos Araújo Carvalhais Fonseca

(Associate Professor with Habilitation)

Co-supervisor: João Pedro Silva Ribeiro (PhD student in Digital Media)

June 2022

Visual Aesthetics in Digital Games: A Comparative Analysis Between Photorealism and Stylized Graphics

Pedro Lopes Negrão de Lima

Master in Multimedia, University of Porto

Aprovado em provas públicas pelo Júri:

Presidente: Professor Doutor Bruno Sérgio Gonçalves Giesteira

(Professor Auxiliar, Faculdade de Belas Artes da Universidade do Porto)

Vogal Externo: Professor Doutor Rodrigo Hernández-Ramírez

(Professor Auxiliar, IADE - Universidade Europeia)

Orientador: Doutor José Miguel Santos Araújo Carvalhais Fonseca

(Professor Auxiliar C/Agregação, Faculdade de Belas Artes da Universidade do Porto)

Abstract

Based on the assumption that digital games often rely on some kind of visual display, this dissertation explores two prevalent visual styles in digital games — photorealistic and stylized graphics — to understand the challenges of incorporating them into the creation of a game. The topic at hand was chosen to be studied through the development of a game prototype that adopt each style, focusing on the exploration of their technological and artistic qualities. The prototype development was structured around the Design Sprint methodology and project management tools to ensure its completion in due time. To lay the foundation for the creation of a game prototype, a review of the literature on digital game appearance and the evolution of both photorealistic and stylized styles was undertaken. The results were recorded so that its findings can enhance knowledge to inform practitioners and designers trying to break into the independent game industry.

Keywords: digital game art, visual style, photorealism, stylized graphics, game design.

Resumo

Partindo do pressuposto de que os jogos digitais muitas vezes dependem de algum tipo de informação disponibilizada visualmente ao jogador, esta dissertação explora dois estilos visuais predominantes em jogos digitais — gráficos fotorrealistas e estilizados — para entender os desafios de incorporá-los na criação de um jogo O tema em questão foi escolhido para ser estudado através do desenvolvimento de um protótipo de jogo que adota cada estilo, focando na exploração de suas qualidades tecnológicas e artísticas. O seu desenvolvimento foi estruturado em torno da metodologia Design Sprint e de ferramentas de gestão de projetos para garantir a sua conclusão atempadamente. A revisão da literatura sobre aparência de jogos digitais e a evolução dos estilos fotorrealista e estilizado foi realizada como fundamentação teórica. Os resultados foram registados para que as descobertas possam aprimorar o conhecimento do campo e informar profissionais e designers que ambicionam entrar na indústria de jogos independentes.

Palavras-chave: arte para jogos digitais, estilo visual, fotorrealismo, gráficos estilizados, design de jogos.

Acknowledgments

Firstly, I want to thank my supervisor Professor Miguel Carvalhais and my co-supervisor Doctoral Student João Pedro Silva Ribeiro for guiding me through the dissertation.

My family, for encouragement, and love. Even being physically far, we are always together.

To my friends that helped me to relieve stress every weekend or so with a healthy amount of cooperative gaming. These sessions not only gave me the motivation to keep going, but they were also highly inspiring times of debate about game design that resonated in this study.

To Daniel Moori for, continuously, helping with, not only technical Unity matters but with how to be a professional game designer. In more than twenty years of friendship you continue to be a great inspiration.

A special thanks to my life companion, Danielle Beduschi, for being always the greatest partner for every occasion, either trying to pass me some of your academic knowledge, on with more important matters. Thank you for your unconditional support and patience, especially when things get rough.

Pedro Lopes Negrão de Lima

Index

Figure List

Table List

1. Introduction

1.1 Context and Relevance

Digital games have come a long way since their invention. Their role in modern society as a consolidated media can be perceived in the size of its industry, and in the impact it has in many spheres of our culture (Horban et al., 2019). Digital games have also paved their way in academia. Its multidisciplinary nature makes it an interesting object of study for different fields. However, with the invigoration of the media, a specific field of study was formed around it, being today an area of research in itself.

Much was done to understand how digital games are structured. Jesse Schell (2015), formulated one of the most widely acknowledged descriptions in the area, dividing a game into four elements: *Mechanics*, *Story*, *Technology*, and *Aesthetics*. Robin Hunicke et al. (2004) presented the MDA (*Mechanics, Dynamics, and Aesthetics*) framework as their structuring endeavor. In both cases, *aesthetics* is recognized as one of the building blocks of a digital game.

Aesthetics in digital games concerns the aspects related to the player experience (Fabito & Cabredo, 2019; Hunicke et al., 2004; Lei et al., 2021; Schell, 2015). It has many facets and touchpoints with philosophy, psychology and arts, to name a few. The term *look and feel* is often used by the game community when discussing aesthetics. It can be an indication of two commonly observed aspects. Game *feel*, as presented in Steve Swink (2008), has to do with the sensation of involvement with a game and encompasses several aesthetic concerns.

On the other hand, a game's look relates to the digital game's graphical style or even the chosen strategies to convey the representation of game elements in a visual manner. This dissertation focuses on aspects related to *look*. However, it will sometimes touch on aspects closer to *feel* as both are building elements of the same game experience.

At first glance, graphics can seem of second-order importance in the player's overall aesthetic experience, but every digital game presents some sort of visual display. Game visuals are the first thing the player contacts with. Sometimes, even before getting in touch with the game, they may see an image or a gameplay video from other media. Once within the game, one of the first tasks of the player is learning to identify different elements displayed onscreen to create an understanding of their function and behavior (Wolf, 2013).

Looking at 2021's list of best-selling games on the *Steam* platform – currently, the biggest online distributor of digital games — of the twelve games listed (Valve Corporation, 2021), eight can be described as making use of the *photorealistic style*. In their list of top new releases of the year, the same tendency can be observed, with the majority of titles utilizing the same style. On the other hand, many recent successful games escape photorealism in a search of the *stylized* look tag. Digital games can use different techniques to create the desired experience, including with regard to its visual qualities.

This research will revolve around digital games visual styles, specifically, the two most commonly used terms to describe its visual style: *photorealistic* and *stylized*. "Style is a very concrete factor: games as developed for, judged on, and grouped by their style, and style also signals the aspirations of the creator" (Juul, 2019, p.50). Through the study of style, one can better help to understand the development of an art form in all its complexity (Bordwell, 1998).

1.2 Problem and Research Objectives

Designers and developers who want to start working in the field of game development face many challenges, especially considering the current moment of high competitiveness in the industry and the saturation of virtual stores – with hundreds of games released daily. In addition to technological innovations, the advancement of the game culture has reached a more diverse audience. Different necessities have emerged from those existing at the beginning of the video game era that needs to be addressed in order to enrich the field.

The problem this dissertation wants to address has to do with one of those challenges: By which parameters — from an artistic and technological perspective — can the visual style of a digital game be designed and developed, and what challenges their implementation presents?

From this problem, some questions are raised:

- What is a visual style?
- How is it defined?
- Once chosen, how to implement it in a game in a cohesive manner?

The intention of this dissertation is to answer these questions and help with the decision-making process of choosing the right visual style. It is expected that its results can contribute to the development of knowledge that can be used in practice and can inform other designers in the process of creating three-dimensional digital games. This dissertation seeks to achieve the following objectives:

Main Objective

- The creation of a prototype that uses photorealistic and stylized graphics to describe and understand the techno-artistic challenges of implementing each of the styles in a solo development scenario to help with the practice of designing this kind of content.

Secondary Objectives

- Describe defining elements of the photorealistic style and stylized visuals;
- Identify implementation feasibility from a techno-artistic standpoint;
- Create documentation of the complete process.

1.3 Methodology

The Design-based Research model, as described by Feng Wang and Michael Hannafin (2005), is defined as a systematic but flexible methodology. It aims to achieve contextually sensitive design principles and theories (Wang & Hannafin, 2005), fitting the purpose of this dissertation.

Design-based Research can follow different approaches in relation to the nature of the research and its connection with its five principles: "(a) pragmatic; (b) grounded; (c) interactive, iterative, and flexible; (d) integrative; and (e) contextual" (Wang & Hannafin, 2005, p.3).

Therefore, this research will begin with a compilation of relevant literature on the topic, following the theory-driven approach of the model. The bibliographic survey was structured around research that had a direct relationship with the concepts and contexts of digital games and that examined digital games as media. The selected bibliography was based on the relevance of the subjects addressed to meet the question and research objectives. ScienceDirect, ACM Digital Library, Scopus and Google Scholar were the scientific research platforms chosen as the primary sources for the survey. The queries were done within a set of parameters: year of publication starting in 2000; keywords utilized were "aesthetics+videogames", "graphics+videogame", "style+videogame", "aesthetics+digital games", "graphics+digital games", "style+digital games".

Due to the nature of the digital games field, other relevant sources were also used in the survey process. The Game Developers Conference (GDC) release each year their conference talks in video format on their website and on their official channel on *YouTube*. The website *GameDeveloper*, formerly known as *Gamasutra*, is a compendium of texts and blog entries done by several prominent figures from the game development industry since 1997. Both GDC's talks and *GameDeveloper's* texts are an important body of information for the field and were consulted in this dissertation as well.

Following the bibliographic survey, a prototype of a game will be created to emulate a real-world setting of a solo developer context. Wang and Hannafin (2005) point to the need to connect the design process and the setting where research is conducted with its results. Thus, the creation of a prototype is an exercise that exemplifies well the work of a solo designer defining the visuals of a game and lead to relevant insights to this creation process. This prototype was created in an iterative process, and many of its basic assumptions may have changed along the course.

Later, the research process, research findings, and changes from the initial plan are documented. More than specified actions to be followed, the findings transcend the immediate problem setting and context to guide designers in both relevant theory and generating new findings (Wang & Hannafin, 2005). As a result, interested game designers or researchers can exam contextual factors or conditions that led to particular effects (Baumgartner & Bell, 2002 apud Wang & Hannafin, 2005).

1.4 Dissertation Structure

In addition to the introduction, this dissertation contains three more chapters. Chapter 2 is a bibliographic survey in which related works are presented. The chapter is divided into three sections: the first deals with the definition of visual style and its elemental characteristics; the second section presents the photorealistic style and its qualities; and the last section discusses the stylized visuals of digital games.

In chapter 3, the creation of a digital game prototype will be described. Game design elements covered in chapter 2 will be utilized to define the prototype qualities and help to define its scope. The emphasis will be on the description of artistic and technological design choices and the subsequent results of those decisions.

Chapter 4 is dedicated to the research findings. The principles and design methods employed in the prototype will be analyzed so that these procedures can likely be proved effective in new settings.

2. Digital Games: Appearance and Visual Style Development

2.1 What is Visual Style in Digital Games?

A digital game is a complex system composed of many layers. The player ends up not having contact with all those layers, such as the endless lines of code that control the game's behavior or even the logic behind every design decision. This computational layer is what Frieder Nake (2018) calls *subface*, an element that works together with the *surface* - the sensitive layer - to form the dual nature of a digital work of art. While the subface is often not noticeable, "hidden in computing equipment, in the cloud" (p. 24), the surface is the part we can access through our senses.

In the case of digital games, the surface layer can be interpreted as the outputted visual and sound information constantly being displayed on the screen, speakers, or another output device. This information will convey all the game visual stimuli through the imagetic representation of the game elements. It is also what will allow the player to access the unfolding game world (Nitsche, 2008) they are about to experience.

All these game elements, their design, and implementation affect how the visual appearance of an individual game ends up being. Aki Järvinen (2009) compares them with architecture, in which the elements and details of a building together constitute its architectonic appearance. Throughout its history, digital games have had different appearances due to the particular way they were created. Factors such as available technology, cultural trends, and the artistic expression of their authors got combined to achieve the desired look. *Style* is the term utilized to define the particular way used by an artist to output his work (Muller, 1979 apud Gee & Dolah, 2018). The term graphical or visual style is often used in the game community to describe digital game visual qualities.

The visual style is not an exclusive concern for digital games. Instead, it is a topic shared with all media with visual content. Let us look for a definition of style in other media, such as photography or film. We can find relatable concerns but also elements specific to that media.

Michael Langford (2000) presents a definition of style in photography by stating that style is difficult to define but recognizable when you see it. He continues by saying that "pictures are characterized by having a mix between the desired mood, treatment, use of color, composition, the proportions of the picture and the subject captured" (p. 12). Although these elements can be read subjectively, they are rooted in technical matters. Adjusting the camera's aperture, for instance, can lead to a darker image, directly influencing its mood. The artist needs to understand and control de camera settings to get the desired look. An analogous behavior will occur in digital games. Game artists must master the tools to create computer-generated imagery (CGI) to produce the desired visual style.

As we jump to film, this relation with technical qualities can be further observed. In the seminal book, *On the History of Film Style*, David Bordwell (1998) defines "film style, in the narrowest sense, to be as a film's systematic and significant use of techniques of the medium" (p.4). Those techniques fall into broad domains: *mise-en-scène* (staging, lighting, performance, and setting), framing, focus, control of color values, and other aspects of cinematography. Some elements have a directed connection with photography and will continue to be important in native digital media such as the digital game. For instance, the camera has a virtual equivalent in the game development environment, so concerns like focus and framing will be inherited and absorbed.

Visual style is easily understood among the digital game's community since, as time passed and game production became more sophisticated, game design evolved into the orchestration of various disciplines. Big studios developing AAA games, as the big-budget games are known, tend to have a department dedicated to the development of all the game's visual elements. This art department is often composed of different professionals, such as concept art developers, 3D artists, environmental artists, technical artists, and art directors. Today we can still find people developing games alone, like in the 1980s when digital games flourished. However, these can be considered a one-person band as they will have to wear many hats to complete their work.

In the field of digital games research, it is not easy to find a definition for visual style as the term is so freely used that its definition appears to be somehow pre-established. Michael Nitsche (2008) describes a concept he calls *presentation,* which can be understood as "the expressive and representational element of 3D video games" (p.67). It is representational because it is responsible for visually displaying the computational subface necessary to create the game world in a comprehensible manner. Nitsche (2008) also calls it expressive, denoting that the elements displayed can have different expressions and assume different visual properties. This expression can also be read as the element that brings this concept closer to a form of artistic expression.

In the visual arts, many researchers still find support to describe visual style as this issue has been debated for much longer. One example estate that variations in style are achieved through the relationship between *form* and the general *quality outlook* - which, in turn, is based on principle elements, techniques, expression and inspiration (Dondis, 1973, Schapiro, 1994 apud

Gee & Dolah, 2018). The use of references from classical arts is proper because it leaves aside some more subjective concerns and focuses on the graphic qualities of the studied work.

Having all these definitions and theories as a bottom plate, this dissertation will try to negotiate a perspective that serves its approach to visual style and game appearance. Thus, the terminology used to describe the visual treatment and overall visual quality outlook of a specific game will be *appearance*. Thus, we can say that the appearance of the game *Mario 64* (1994) can be described as blocky, three-dimensional (3D) with cartoony features. The term *style* or *visual style* will be used interchangeably to describe a collection of games with a similar appearance. In this manner, the early photorealism of AAA games from the mid-1990s can be named a visual style. Therefore, we can say that an individual game has an appearance. According to it, the game belongs to a specific style (Järvinen, 2009).

2.1.1 Defining elements of game appearance

With the support of other media forms, we can define game visual style, but which elements work together to compose the appearance of a particular game? Do digital games share some elements with other media, or do their idiosyncrasies make them unique? Digital games have adopted or even directly inherited some elements from older media – like the camera properties presented earlier – but being one of the first media to be natively digital rather than being transferred to digital format, many of their aspects are new.

In an attempt to conceptualize and categorize audiovisual style in digital games, Järvinen (2009) indicates three fundamental elements for every game: space/environment, objects, and symbols. The author points to the first – space/environment - as the most prominent element, a fact that aligns with a discursive tradition on space in digital games explored by many other authors.

Espen Aarseth (2001) affirms that computer games are essentially concerned with spatial representation. Having a foundation in Leirfall and Lefebvre, authors who have focused previously on the theme of "virtual space" and its relation with real space, he proposes that "the representation of space in digital games is a reductive operation leading to a representation of space that is not in itself spatial, but symbolic and rule based" (2001, p. 163). A game is an artifact based on rules by definition. The generated spatial representation enables the game experience. It has a reductionist nature, as it intends to achieve the gameplay objectives, showing only what is intended for the game in the most advantageous manner. "In real space, there would be no automatic rules, only social rules and physical laws" (Aarseth, 2001, p. 163).

This reductionism can be observed in contemporary games that display their visual elements in two dimensions (2D) instead of 3D. At first glance, using a 3D environment may always seem preferable, as 3D graphics allow another dimension of representation and expression (Nitsche, 2008). It opens up the possibilities for three-dimensional space navigation and exploration, but if the desired gameplay does not account for these features, creating a 3D space would only work against the game. In a case like this, a 2D representation would be preferable as it may convey the spatiality of the game more directly, serving better to the overall experience.

Presenting another approach to the topic, Mark Wolf (2001) introduces one of the first attempts to categorize spatial structures in digital games. By indicating what was already done and the potentialities of using space in games, the author defines the digital game itself. At the beginning of the 2000s, it was not very clear exactly what a digital game was, and the efforts of researchers at the time revolved around defining the boundaries of the field. Nitsche (2008) blames the term *multimedia* for "an uncritical blending of different presentation formats into 'virtual spaces'" (2008, p. 5), pointing out that it can be an oversimplification to assume that 2D worlds, interactive cinema, and 3D games to share the same concept of space.

Wolf (2001) enumerates eleven types of structures, which can appear individually or combined in the same game. Each use of space can be more appropriate to a specific kind of game (Wolf, 2001). He starts with what he calls "no visual space; all text-based" for 1980s text adventure games. Passing through many 2D possibilities that occurred at that time, Wolf (2001) and ends his list with "interactive 3D environments". Besides describing the structure and giving examples of games that make use of it, his record also establishes some relations to other media such as books, in the case of text-based games, and film. By today's standards, the text may seem outdated in its categorization as many of those points are not used anymore or seem confusing as it mixes elements that work separately today. However, as the first of its kind, the analysis presented is of immense value. It made it possible to conduct further work on the subject, helping to ground the field of digital games as an academic discipline.

Almost a decade later, Järvinen (2009), focusing on the same topic, subdivides space into elements that have a more precise description. This division makes it easier to understand its use as it helps create terminology to be used by developers, journalists, and players. The elements he proposes are *Dimension*, *Point of perception*, *Soundscape*, *Senso-motorism*, and *Visual outlook*.

2.1.1.1 Dimension

In mathematics, the dimension of an object is defined as the number of coordinates needed to specify any point within it. (Weisstein, n.d.) We would need only one coordinate to specify a point on a line, while a point on a rectangle would need two coordinates - width and height – to be described. In digital games, dimension is used to define the nature of the simulated space. Text-based, two-dimensional, and 3D are some of the most common options. However, game designers are always experimenting with new possibilities. Besides having an impact on the mathematical simulations necessary to create the game world, dimension affects the player's experience, influencing how they perceive space (Järvinen, 2009).

Nowadays, there is a predilection to 3D environments as the barrier once raised by technology constraints fell, and its implementation has become increasingly ordinary. Even though most 3D worlds in digital games are experienced via a 2D screen – an exception would be games that use Virtual Reality Displays to simulate the 3D world – some specific qualities set it aside from other formats (Nitsche, 2008). 3D game environments often create a stronger or at least a more complex sense of place than 2D environments (Järvinen, 2009).

It would be naïve to abandon every other possibility on dimension form, such as the classic 2D, as they have unique ways to convey visual information. As seen before in Aarseth (2001), the element of dimension works at its best when used to support gameplay and, also, to convey narrative.

2.1.1.2 Point of Perception

The Point of Perception (PoP) is the element that describes the player's point of view of the space. A possible analogy would be the position of the camera, and microphone, from which the game is viewed and heard. "It can be compared to the dominating view discussed in theories of narratives and narration, but it is not quite the same thing, because of the highly interactive and non-narrative nature of most games" (Järvinen, 2009, p.116).

Digital games often explore the use of different PoP in order to find the option that best fits the intended gaming experience. Using the main character as a focus, the point of perception can simulate their view. However, it can also be dragged out, framing a part of the character or his entire figure. The first originates the *First-person view* and can be found in shooting games, creating the *First-person shooters* (FPS) game genre.

The second puts us on the *Third-person view* gamut, in which many common camera positions have a denomination. If the camera is closer to the character, it is referred to as *Over the Shoulder Camera* or, simply, *Shoulder Camera.* Dragging the camera further away of the main character creates the *Bird's eye view* and even back, to a *God view* in games in which a lone character is less important than the management of a group of characters, buildings and resources like in a real-time strategy (RTS) game.

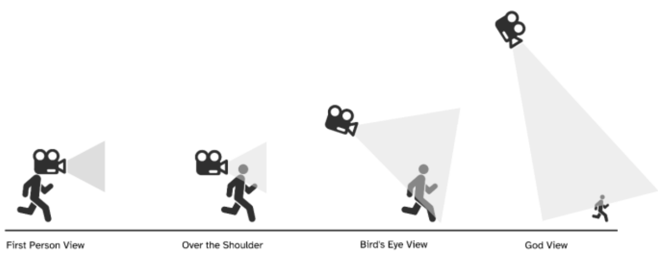

Figure 1 - Different Points of Perception that can be found in digital games.

It is essential to know that PoP can be dynamic and controllable in a digital game. Suppose it makes sense to the pretended experience or to create a better understanding of the environment. In that case, the player can control the camera position, zoom, and focus (Järvinen, 2009). Other than controlled, it can also be entirely changed, in some cases, as the player can choose to play in a first-person view or an "over the shoulder" camera as it suits them better. A game can even have two PoP working at the same time. Two simultaneous PoP is a feature observed in Aarseth (2001) and Wolf (2001) and called by the latter as "represented or 'mapped' space". This feature often appears as a 2D map representation of the environment displayed as a separated view or an overlaid element on top of the main PoP. It serves as a guide when the environment is too large and complex to be easily understood. These maps can have a gameplay function besides being schematics of the environment. They often work as a compass or as a radar to display the position of enemies in the level.

2.1.1.3 Visual Outlook

While the elements of dimension and PoP compose the structure of the game world space, the element of visual outlook deals with the appearance of the content displayed. Järvinen (2009) reduces this element to the "so-called textures and polygons" but this topic deserves a deeper exploration, as visual appearance is a primary concern in this dissertation.

Järvinen (2009) also mentions that the visual outlook can be derived from another media artifact. This is better seen in games that are part of a franchise or Intellectual Property (IP), such as the ones owned by big companies like Disney that launch games to accompany the success of their movies. However, this is also something that can also be found when a digital game wants to borrow a familiar aspect, defined on another media to its own, as with classic narrative genres. For instance, if a game opts to be a *Noir* experience, it can adopt some familiar elements, or "audiovisual motifs" (Järvinen, 2009), from other *Noir* works, as books and films, to convey the look and feel of the genre. This point of contact of visual style and narrative genre will be discussed further in the following section.

2.1.1.4 Soundscape

The Soundscape element is responsible for the sonic qualities of the digital game. It is the sum of diegetic and non-diegetic sounds. Diegetic sounds are essential to convey spatial aspects and amplify the environmental sense experienced (Järvinen, 2009). Non-diegetic sounds, on the other hand, are often used in the same manner it is used in film: to create the desired atmosphere and help with essential plot moments of expectation and suspense.

The sonic dimension of a digital game is as important to the game experience as it is complex. Being sound design a field on its own, it will be not explored in depth by this dissertation.

2.1.1.5 Senso-motorism

This element of spatial perception accounts for the interface system the player uses to interact with the game environment (Järvinen, 2009). There is a difference between the interface found on the game menus and starting screen and the interface used to navigate the game world. The first, having more in common with web and software design, should conform to the usability rules. The second one, the interface with the actual game, works differently because it can require some sensory difficulty to make the game rewarding and enjoyable (Järvinen, 2009).

In defining game appearance, Dimension and Point o Perception are fundamental. Visual output, in these terms, is too broad but can be broken down, as we will see further. The element of Soundscape is as vital as the visual output to the audiovisual experience of the player but is outside the scope of this research. The same could be said about senso-motorism as both will not be a part of the main body of study presented in this dissertation.

2.1.2 Form, Motion, Surfaces & Light

A digital game art director's concern is about the look and feel that better suits the proposed game design. To have a more concrete vision of their idea, they will ask the concept artist to materialize their intentions in illustrations. Considering that everything goes well and the concept is approved, the illustrations will be passed to the designers and artists, who will create the elements used in the game itself. Being this a 3D game - as most of them are nowadays - these professionals will be 3D artists, 3D environmental artists, 3D character artists, Visual effects (VFX) artist, among others. These professionals will work with highly specialized software to create the three-dimensional elements, often called assets, to bring the game to life.

This, of course, is an oversimplification of the process of creation of visual content for digital games today. However, the above description serves the purpose of introducing the game artist's work of creating the content that will give form to the game experience. The software used is based on the rationality created in the computer graphics field and all of its advancements over the years. McLaughlin et al. (2010) point out that computer-generated imagery (CGI) has grown in three main fields: *Form*, *Motion*, and *Surface & Light*. This will serve as a guide to understanding the fundamental elements that compose game asset creation and, ultimately, define a game's appearance.

Resorting, once again, to the field of the arts, the term *form* can be understood as the visual element that is three-dimensional and encloses a volume, having length, width, and height (Marder, 2020). For its 2D counterpart, the term would be *shape*, but often these terms have their meaning mixed, especially when artists talk about defining the geometry or the silhouette of an important component of the art piece.

From a 3D asset creation perspective, form concerns questions related to modeling 3D objects. A 3D model is the mathematical representation of any three-dimensional object. A user-friendly layer generally hides this math layer in the appropriate modeling software, such as

Autodesk Maya, Blender, or Pixologic Zbrush. These software are industry standard in creating 3d models for games. All of them work to make creating models more practical for a game artist.

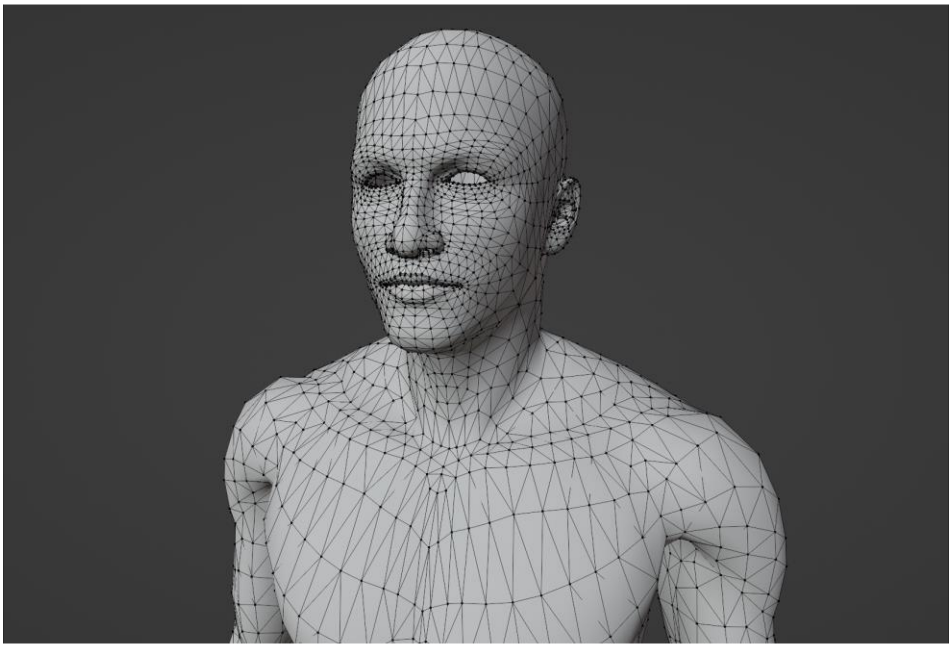

Figure 2 A 3D mesh composed of triangular polygons.

In developing a game asset, modeling has to do with the shape of the object, its geometry, proportions, and silhouette. The 3D shape is created on the geometry of triangles, which are constructed by lines built from points (Peddie, 2013). The triangles, often called polygons, are combined in a mesh to obtain the complex forms of a character, a building, and other objects. Models with a high number of polygons in their mesh probably represent very detailed objects, made to be viewed close to the camera. If a model has a low number of polygons, it can appear as blocky or choppy, as it may lose some of the polygons that define its silhouette. On the other hand, models with high polygon count are more demanding in hardware processing power. In contrast, low poly count ones are easier handled, making them preferable to be used in a game even today when processing power is higher than ever was before.

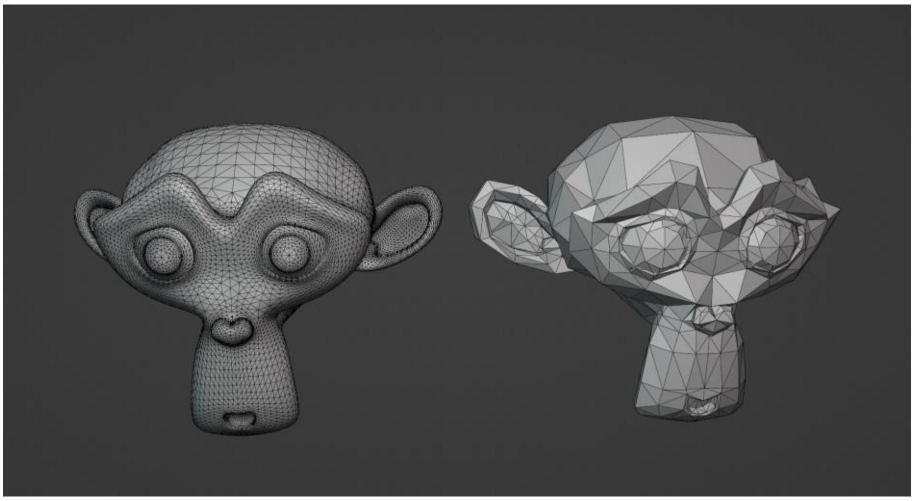

Figure 3 - The right model is composed of fewer polygons, so its shape seems blocky compared to the left high poly version.

Motion has to do with the animation of the models. There are many methods to animate a 3D model, and *rigging* is one of the most used techniques (Peddie, 2013). It consists in using a skeleton to control the movement of the model's mesh. Each bone in the skeleton is in charge of managing a particular area of the mesh, so whenever a bone moves, that portion of the mesh moves along with it.

Figure 4 - Example of a model with internal animation rig.

Surface and light are more technically called rendering. It deals with questions about how light will interact with the surface of the modeled object to make it discernible to the viewer. Models are not technically graphics until they are somehow visually displayed. Even modelling software has to implement some rendering method so the model can be visualized as it is created and edited.

Rendering is a vast topic that surrounds the computer graphics field intensely. Simulation of light behavior as physically accurate as possible, reflection of different kinds of material, and post-effect application on rendered images are some of the issues that the field of computer graphics has addressed (Peddie, 2013). Concerning digital games' visual appearance, it is important to know that contemporary rendering can be divided into two elements: *Materials* and *Illumination*.

Materials are composed of 2D graphics, often called textures, that contain information about a model's superficial characteristics. If, for example, one has modeled a car and wants it to have a red metallic surface appearance, one can generate a red color texture and a metallic property texture to create the material for the car. Utilizing 2D textures in 3D models is the most computational cost-effective way to add superficial information, creating more exciting and expressive results (Peddie, 2013).

The method used to illuminate the game environment is the last element. Light will play a significant role in how the digital game's final appearance. In a fully digital environment light have to be created by simulation. The simulated nature of the light on the digital environment can follow distinct methods to be calculated, leading to unique results. Some methods are more physically accurate, resulting in an illumination that resembles more the natural effects of light. Other models are non-physical based or are abstractions developed to facilitate the calculations needed to simulate the light behavior on the environment. These models will show less accurate results in simulating real-world light at the expense of being easier to compute.

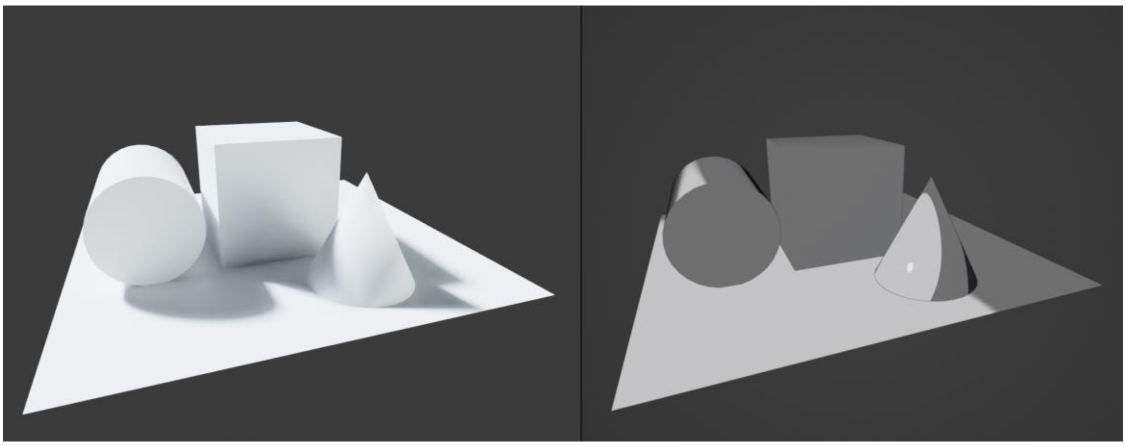

Figure 5 - Same models with different rendering settings lead to different outcomes.

The manipulation of these three elements – the model, animation, and rendering – can generate many unique visual outlooks and are fundamental in defining a game's appearance.

Many of these combinations had standards copied and followed by many art departments, creating the visual styles we know today. The most common one is the Photorealistic style that follows the most physically based and grounded on the real-world methods of simulation as possible. Every other option ends up being grouped under the umbrella term of *Stylized* visuals. Both visual styles will be discussed further in the following sections.

2.1.3 Characters representation x Environment representation

Many games depend on player characters, "which represent the player's influence and effect on the game's diegetic world" (Wolf, 2013, p.50). Player characters can be of the *implied type* when they are not visible on screen, as in an FPS game in which the player assumes the PoP of the character or in an RTS, in which the player views the environment from above and their actions are represented by the moving cursor manipulating the game objects on the screen.

When the player-character has a visible graphical representation, an avatar, they are called *surrogate-based* (Wolf, 2013). In early games, these avatars had to assume elementary forms to be displayed, as computational power was scarce. Wolf (2013) exemplifies this with Atari's Football (1976), which had to use "X"s and "O"s to represent the players instead of human figures. Nowadays, the anthropomorphic character has become the standard option for player-character, as computation costs become less of an issue, and the relationship a player has with the human figure is more meaningful than with the abstract forms from the early games.

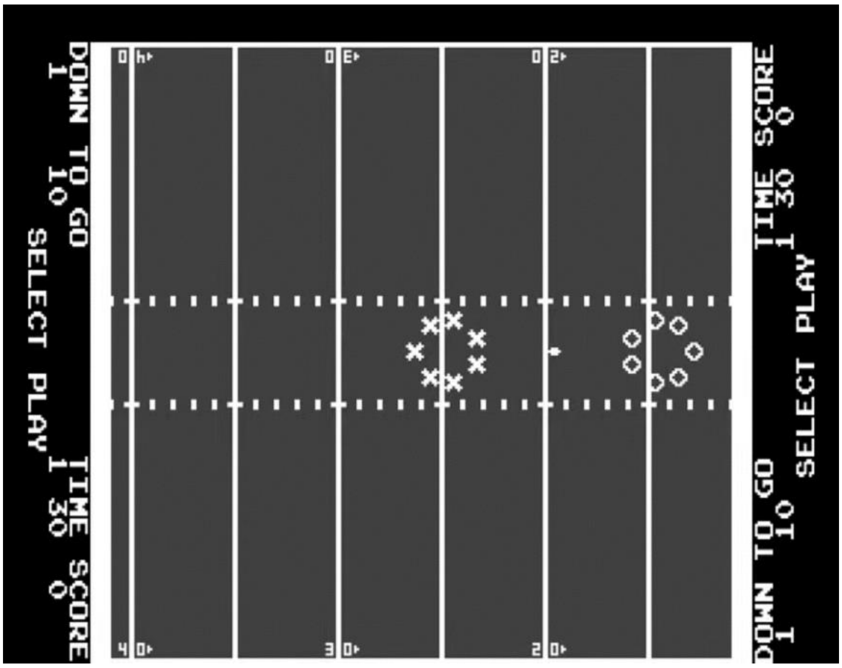

Figure 6 – Old Classic Retro Gaming (2014). Screen capture depicting the game *Football*.

However, to amplify the player's identification with its avatar, it is possible to apply methods already explored in other media, such as the simplification of forms in character appearance. In comic books, the use of simpler forms to represent humans, creating cartoonish-looking characters, is often used when the author wants to create a stronger identification of the reader with the comics characters. "The cartoon is a vacuum into which our identity and awareness are pulled, an empty shell that we inhabit which enables us to travel to another realm. We don't just observe the cartoon, we become it!" (McCloud, 1993 apud Wolf, 2013).

As in comic books, characters and environments can have different appearances to achieve different objectives. At the same time, as cartoony characters can be more relatable, Scott McCloud (1993) points out that a more realistic environment can feel more grounded and recognizable. Thus, a digital game can make use of this concept and combine characters with cartoony visuals to a realistic environment representations to shape the experience in which the player is allowed to "mask themselves in a character and safely enter a sensually stimulating world." (1993, p.43)

A clear example that makes use of this combination is the game *DokeV* (2022). The game mixes a highly realistic appearance for its environment while utilizing an anime-like look for its character designs. *Mario Odyssey* (2017) also makes use of this combination in some parts of the game. In this case, the stylization, traditional within the series of games, is kept for the most part, but a realistic environment level is employed. It is possible even to say that *Final Fantasy VII* (1997) employs this technique, though maybe not for the same reasons: the developers had to apply some kind of stylization to the characters that had to move in real-time while the environment could be pre-rendered in a more realistic fashion due to the limited computational power at the time.

Figure 7 - *DokeV* (2022), *Mario Odyssey* (2017), *Final Fantasy VII* (1997), respectively.

We can extrapolate this separation to the previously presented elements described – the model, animation, and rendering. These also can assume different visual outlooks in the same digital game, leading to exciting results. Creative Shrimp (2021) exemplifies this by analyzing an image and breaking it down into render style and design style. While the rendering style is highly photorealistic, the car design leans towards a more stylized representation, shifting the perception of the image towards the stylized side.

2.1.4 Visual style and the narrative genres

Despite their similarities, digital games can be considered as non-narrative products, as several researchers and scholars argued throughout the game's story as an academic field of study (Frasca, 2003; Juul, 2001; Ryan, 2001). From time to time, games lean on traditional narrative media as they are much older, well established, and familiar. The use of narrative genres – such as horror, western, and sci-fi – is one of those times.

These well-known genres carry conventions that, among other things, can affect the appearance of a game (Järvinen, 2009). Games can share many of the same defining elements - *Dimension*, *Point of perception*, *Soundscape*, *Senso-motorism*, and *Visual outlook* – and even their game mechanics but still have a unique look and feel. Järvinen (2009) exemplifies this question with like *Silent Hill 2* (2001), *Grand Theft Auto III* (2001), *Jet Set Radio Future* (2002), and *Ico* (2001).

Figure 8 - *Silent Hill 2* (2001), *Grand Theft Auto III* (2001), *Jet Set Radio Future* (2002), and *Ico* (2001), respectively.

Genre in digital games is a topic of discussion (Järvinen, 2009; Wolf, 2001). It is a matter that often gets entangled with the narrative genre, especially when players try to label a game they just played but lack the vocabulary to do it in an informed way. Game sellers and specialized journalism are adopting more a format that adds interaction mechanics plus audiovisual appearance as genre descriptors. This can be perceived in terms such as "survival-horror" games or "fantasy-rpg". This solution seems more appropriate as it embraces narrative and game mechanics elements, creating a more complete genre descriptor.

The present work intends to analyze the visual style, despite the influence of the narrative genre that games can receive by using the fundamental elements described in this chapter. Leaning once again on film and film theory, Bordwell (1998) places his On the History of Film Style book under the umbrella category of aesthetic history, together with the history of film form and the history of the genre (p.4). If we create a parallel with this system, this dissertation would be framed in the analysis of style, under the umbrella category of digital game aesthetics, and genre could be seen as another sub-category under the same umbrella.

2.2 Photorealistic Style

To make a computer generate images was something sought after from its early development. They have done it since the 1950s with the attribution of the term *Computer Graphics* (CG) given by William Fetter, a graphic designer for Boeing at the time (Peddie, 2013). Its commercial application started in manufacturing companies in the form of CAD (Computer-Aided Design) software that initially displayed 2D lines and shapes to help engineers develop technical drawings. At the same time, computer graphics were used in universities for scientific research. The construction of 3D models of molecules, the visualization of fluid simulations and other physical and mathematical phenomena were themes explored by Computer Graphics departments in many organizations in the second half of the 20th century (Peddie, 2013).

During all this time, a unifying theme that guided some of the research done was the progressive improvement of the graphics' visual quality. Eihachiro Nakamae and Katsumi Tadamura (1995) point to crucial milestones in the history of computer graphics while listing the main achievements to reach what they call *photorealism* in computer graphics. The extensive list shows how the concerns changed from more straightforward issues – like adding color to a 3D model – to more complex ones as time passed. In the 1980s, computational power and the maturity of the field of research provided the means for developing physically-based approaches to rendering (Pharr et al., 2004). The search for better visual quality found a way forward in the physics of light behavior in the real world.

For Lev Manovich (2002), progression towards realism was always presented in trade publications and research papers, the history of technological innovation, research, and media. The author points to SIGGRAPH (Special Interest Group on Computer Graphics and Interactive Techniques), the most renewed convention on computer graphics that started in 1974, as a source of a common direction towards synthetic realism. The industry's interest in the projects developed by engineers and researchers was growing as new possibilities in implementation were demonstrated.

The military industry was particularly interested in what was being developed. The intended use was mainly on training simulation for pilots. (Beekman, 2014) The film industry was another interested in SIGGRAPH's material to reduce production costs (Manovich, 2002). The realism created by computer graphics would affect these fields and would be affected by them as well. Many CG engineers ended up working with game development. In digital games, the thematic of war is continuously explored, and film aesthetics is often borrowed.

But while Manovich used the term "realism" to describe the nature of the work shown in SIGGRAPH, he made it clear that what was achieved was not realism per se, but only photorealism, "the ability to fake not our perceptual and bodily experience of reality but only its photographic image" (Manovich, 2002, p.181). In the context of digital games, this distinction is essential to understand photorealism as a stylistic choice rather than the mechanism that promotes realism. Although its use can contribute to the perceived realism of the game as a whole, photorealism has to do with imagistic qualities.

Amidst digital game enthusiasts and developers, both terms get confused as they are commonly used interchangeably. However, a game must cover more elements than just its visual style to convey realism. Wanners Ribbens (2013) proposes a model to structure perceived realism in games. Freedom, perceptual penetration, social realism, authenticity and character involvement are analyzed in the model. This model serves as an indication that the issue of realism goes beyond the appearance of the game.

It is hard to imagine that the purpose of using synthetic photorealistic images is different from improving the sense of realism of the media artifact that employs this kind of style. However, detaching the real-like quality of a photorealistic image from all the other elements that convey realism lets the term *photoreal* be applied in situations that would not otherwise be used. We can say that a fantastic creature, like a dragon, can have a photorealistic appearance even if there are no dragons in the real world. It is a matter of creating the illusion of life likeness that matters most.

Järvinen (2009) calls this fictional photorealism, or "second-order realism", in which an element can be perceived as something different from what it is for a determined purpose. For instance, the floor can be perceived as lava in children's play, and kids can only step on the furniture to reach their destination. Once the game ends, the floor returns to being perceived as a regular floor. The dragon can be perceived as photorealistic in a game, although there are no dragons out of that game world.

Photorealistic images aim to be indistinguishable from photographs. However, as Pharr et al. (2004) point out, the term "indistinguishable" can be understood in a variety of ways depending on the observer's perception. One can perceive an image as a photograph. At the same time, another, with a more trained eye, can see the details that tell that the image was computer-generated.

Figure 9- Scene from the trailer of *Clifford the Big Red Dog* (Paramount Pictures, 2021).

In the image above, the dog element is clearly perceived as computer-generated imagery (CGI). Its model is well constructed, its hair texture is very believable, and the lighting and rendering are well integrated into the shot. What breaks the sense of realism is the size and color in comparison to the real-world elements that surround it. Sometimes the perception of realism can be broken, even with a highly photorealistic render but with a design element that does not fit the image content (Creative Shrimp, 2021).

2.2.1 Digital Games and Photorealism

When it is interesting for a movie to integrate CGI with real-world shots in a photorealistic way, the CGI should be based on real-world conditions. Each shot has its specific light conditions, to name one, that the built-in CGI element must adapt to be interpreted as indistinguishable.

Digital games bring different challenges to photorealism. Having its world fully computer-generated, photorealism becomes, firstly, about technology to achieve the photoreal look. What are considered photorealistic changes with the improvements in CG technology and society's perception of these improvements? More advanced algorithms to produce new effects are constantly in development. The effects possible with older algorithms no longer signal the *state of the art,* as they have become available to everyone else in the field (Manovich, 2002).

The game *Metal Gear Solid III* (2005) was praised for, among other qualities, having "beautiful graphics" (Lewis, 2004). Although the definition of beauty can vary, it is possible to determine what was perhaps meant by reading a review from the time the game was published.

> "(…) it still looks superb by any standard. Impressive little details abound, such as how gunfire causes the dense jungle underbrush to violently sway and tatter,

> how blood soaks through Snake's uniform when he's injured, and how the jungle's various indigenous creatures move just as fluidly and realistically as the game's human characters." (Kasavin, 2004)

Today, *Metal Gear Solid III* can still be seen as having beautiful graphics by some people, but the photorealism described would not be perceived in the same way. A game needs to use the most cutting-edge technology to be perceived as photorealistic or it can miss the style entirely. As time goes on and emergent technologies that can produce more faithful graphics appear, the game starts to look out of date due to the technologies that made it photorealistic in the first time.

Figure 10 - (SourceSpy91, 2016) screen capture depicting *Metal Gear Solid III*.

It is also important to point out that although photorealism is tightly related to technology, there is also an economic element connected to it, primarily in terms of computational costs. Adam Finkelstein (2020) points to how it is expensive to create photorealistic games as high polygon models and physically based renders demand a lot of processing power from the computer. It also demands more intense labor from the artist to create detailed models, rising production costs.

Even with these economic constraints, the photorealistic style has many representatives spread throughout the history of digital games. Nitsche (2008) points to a fetishization of some game developers, who seem to compete for the most advanced visuals. This competition seems to happen mainly in the AAA games sector. Games tend to rely on the repetition of a well-worn format, leaving its realistic visuals – based on the state of the art in CG – to be their only new element, as pointed by Andrew Darley (2002).

A notable case is the ecosystem created by Epic Games. They offer their Unreal Engine, a software used to facilitate game creation and Quixel Megascans, a free photorealistic library of assets for in-engine use. Both are free of charge until the published product makes its first 1 million dollars of gross revenue (Epic Games, n.d.). Together with other well-integrated third-party software, Epic Games created a different scenario than what existed a decade ago. Small and independent studios now have access to visuals that were only available to big studios and publishers. The impact on the game development culture of initiatives such as this one is something to be observed in the near future.

2.2.2 Uncanny Valley

Considering the resources that exist today, it is still not trivial to make photorealistic games. It is hard to fool the human perception regarding the appearance of natural elements, primarily when representing the human figure. In the 1970s, in the field of robotics, Mori et al. (2012)[1] introduced the hypothesis that a robot with human likeness would be perceived as more agreeable, until its appearance got too realistic that any subtle imperfections would be disturbing to the observer. The Uncanny Valley, as it is called, became a well-known hypothesis in other fields of knowledge as its application could be adapted to different situations. CG uses this terminology to refer to the effect caused by virtual human figures that unintentionally cause the same disturbing effects to the observer (McDonnell et al., 2012).

In the case of CGI, strangeness has been shown to arise when abnormal features become apparent to highly realistic characters (Seyama and Nagayama 2007; Burleigh et al., 2013, apud Zell et al., 2015). These abnormal features can occur in form, render and animation. When the geometry lacks information, the model can be perceived as uncanny. The same happens with an unrealistic texture or a faulty animation.

[1]The original text, written in 1970 was in Japanese. It saw many translations over the decades, but the version chosen in this dissertation is the first official translation, published in 2012.

Figure 11 - Image taken from *Mass Effect: Andromeda* (2017)

A case in which the uncanniness became massively reported in the specialized gaming media happened in *Mass Effect: Andromeda* (2017). This instance of the *Mass Effect* series suffered from texture and animation problems that made the characters look creepy, falling right under the Uncanny Valley effect. After the game's launch, the developers had to fix those problems due to the criticism circulating in the media.

Working with a photorealistic style can be considered risky. It can produce a negative reaction when the result misses its mark (Geller 2008; Levi 2004 apud McDonnell et al., 2012). Other than that, digital games that use this style are bound to be associated with the time they were created due to the use of technology that, eventually, will be surpassed. Game designers are aware of this, as many of them avoid photorealism. To create something that holds better to the passage of time, to avoid UV effects or to express a different artistic expression, developers resort to some sort of stylization in the visual style of their games.

2.3 Stylized Visuals

In graphical terms, there are many ways to represent an object. Even when considering something ubiquitous as a drawing, there are many possibilities in technique, material and medium that can lead to many different results. Some matters an artist has in mind before – and some even while - drawing a figure concerns the type of drawing instrument to utilize, the type of support to match the instrument; what type of linework to use: what kind of shading, or even if it is going to have any shading at all. These decisions end up impacting the final result, regardless of the subject portrayed.

In digital games, developers took a while until they could worry about artistic possibilities for their games. In the beginning, technological restraints were immense, and making a flickering arrow move across the screen was already a feat. Over time, technological constraints eased but remained a concern. Developers have to find creative solutions to bring their idea to life. Many of these solutions have an impact on the appearance of their games.

Currently, using 8-bit pixels or 3D low poly is a matter of choice — whether artistic expression, ease of finding ready-made kits on the internet or even for financial reasons, since the investment tends to lower with this type of graphic. However, these represent thresholds of a specific technology existing in a specific historical time. Inspired by the visual styles born in those moments, today's digital artists mimic some of its most important features to create many of the *stylized* appearances found in digital games (Creative Shrimp, 2021).

Not only the past of digital games inspires today's game art. Other media can also be a source of inspiration. Simulating traditional techniques like painting or drawing are sought after by a branch of Computer Graphics called Non-photorealistic Rendering (NPR) (Finkelstein, 2020). This method was implemented in games many times and in many ways, but the most evident case is the use of *cell shading*. Also called *toon-shading,* it consists of non-photorealistic rendering designed to make 3D forms appear flat by using less shading color instead of a gradient or tints and shades (Finkelstein, 2020).

Figure 12 - Comparison between a gradient shaded and a toon-shaded model.

One could point out that games that resort to this kind of style are looking to the past to create something new. This claim supports the argument raised by Bolter and Grusin (apud Juul, 2019), which declares that new media tend to *remediate* earlier media forms. These games try to remediate the form of analogic art and even the style of early digital games to communicate what they are and to represent the beliefs of their creators - as it will be discussed later in this section.

Current digital media also can inspire digital game art, with a good example being digital animation films. Sharing computer graphics as a core element of technology, digital animation had many of the same constraints to overcome. However, without the concern of creating real-time interactive imagery, it could focus on other matters. The work was an effort to continue the tradition of cartoon that stretches back to 1930s Disney animations (Darley, 2002).

Different visual styles can grow from these many influences game artists receive – either from references from the past or a film they just watched on the TV. The adopted jargon commonly used in the game community is *stylized* to refer to games that stray from the photorealistic look. It does not refer to a single style. However, it is an umbrella term for the full range of possibilities outside photorealism. It can grow out of all the references mentioned above. Although this does not aim to be an exhaustive list of references, the next few sections will look at some of the key factors that lead to stylized graphics commonly used in digital games.

2.3.1 Abstractionism in early digital games

Digital games had severe restrictions concerning their graphic representation in their earlier forms (Wolf, 2013). *Spacewar* (1961), a digital game pioneer, was created for the PDP-1, a computer with no microprocessor that ran on transistors and diodes and occupied the space of a large server rack. The computer had a cathode-ray tube (CRT) phosphor screen capable of creating one-color images within its 512 points of resolution along each axis (Digital Equipment Corporation, 1960). *Spacewar* (1961) consisted, visually, of the representation of two spaceships, a “burning sun in the middle of the screen”, and a star field in the background (Fleming, 2007). All with the same blue-green color and leaving a photonic drag mark whenever an element moved on the screen.

Figure 13 – *Spacewar*, 1961 (CuriousMarc, 2017).

As technological constraints would not disappear overnight, early games still relied on developers' ingenuity to deliver their visual content. When talking about early digital games, Darley (2002, p.26) observed that those games' imagery was "highly schematic, with flat geometrical figures composed of dots or vector lines". The graphic resources of those games were so limited that they had to adopt a somewhat abstract appearance (Wolf, 2013).

The term *abstract* above refers to abstract art, in which the subject of the work is not the depiction of a real-world element but the art piece in itself (Gersh-Nesic, 2019). As McCloud (1993, p.51) puts it, abstraction is "where shapes, lines and colors can be themselves and not pretend otherwise". In the context of digital games, the term is used to describe game elements that do not have a graphical representation based on their real-world counterparts but use simpler non-figurative forms. Take *Pong* (1972) as an example. The game recreates a ping-pong game, but the rackets are reduced to rectangles, the ball is shown as a moving pixel, and the table becomes a flat two-dimensional background. The same principles are used in the previously mentioned *Football* (1976), where "X"s and "O"s are used to depict players on the screen.

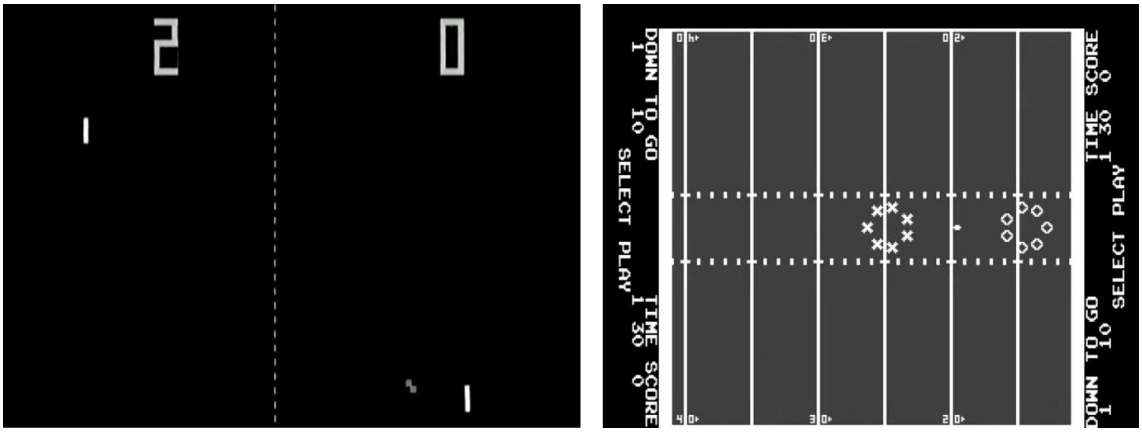

Figure 14 - On the left is a screenshot of *Pong* (1972). On the right, one from *Football* (1976)

It is important to be aware that despite having an abstract appearance, the game elements remain visual markers representative of some aspect of the game. Going back to *Pong* (1972), even with its abstracted look, the game represents a ping-pong match (Wolf, 2013). Abstract here solely describes the appearance of a game and its elements.

The technological restrictions intrinsic to these early games hold back the use of representational elements (Wolf, 2013). The amount of pixel per asset, use of color, and sound were also limited. "These programming restrictions, and others, account for much of the style, graphics, and gameplay of the early Atari games" (p.54). Among many things, Atari was the company that produced the *Atari 2600*, the console that brought digital games into the home environment and became an icon for digital games of the 1980s.

During the same decade, the industry flourished commercially, and digital games began to adopt conventions from other media (Wolf, 2013). Purely abstract games gave way to more representative ones, even though games still retained a degree of abstraction. "As graphics improved and became more representational, abstraction became more of an artistic choice instead of merely a technical default" (p. 58).

Today, it is possible to find new games that make use of abstract elements, especially in the category of casual games. Described as games designed to be played in short sessions, it encompasses a mix of several genres, such as puzzle, words, card games, time management and social games (Aubrey, 2013).

Jasper Juul (2010) defends that the appeal of casual games comes from its distancing from three-dimensional space representation, making it easier to be integrated into the player's life. Going back to *Pong* (1972), a game with such simple graphic elements is very fast in communicating what is it about. Casual games' adoption of abstract elements can make them easier to understand and allow for the deliberately short game sessions common in this category, but the confirmation of this hypothesis requires deeper research which will not be accomplished in this dissertation.

2.3.2 Cartoon tradition, Digital Animation and Games

As Manovich (2002) points out, it is possible to observe CG and the film industry relationship by examining SIGGRAPH's research in the 1980s and beyond. An example given is a paper published in 1987 describing *REYS*, an image rendering software adopted by the company Pixar to compute a feature-length film in commercially viable time and quality (Manovich, 2002).

Both parties would take advantage of this relationship. Computer Graphics would find an industry of interest to invest in the area, thus ensuring that new research could develop advances and ensure further growth. While, for the film industry, this would allow the introduction of a "new degree of expressive 'freedom' to the established media of photography and cinema: one that would maintain the look of photography whilst cutting it loose from its referential ties" (Darley, 2002, p.17). This concept would eventually materialize in the form of Visual Effects (VFX) for *live-action* movies, but this would also open possibilities for animation films as well.

Animation film, in this matter, refers to films fully created by synthetic imagery. Computer animation, in which films are created by digital means (Darley, 2002), was made possible by the technology developed in the Computer Graphics field and displayed in SIGGRAPH conferences throughout the years. But, while the technological end was being covered by engineers and developers, a film still needs to attend to concerns about form, aesthetics and narrative - to cite a few - in order to deliver an acceptable result (Whalen, 2019).

Pixar, for instance, a computer animation studio renewed by the quality — in appearance as in content — of their animation films, saw its first commercial success only after making a partnership with Walt Disney Studios (Cook, 2018). Released in 1995, the result of this partnership was *Toy Story*, the first computer-animated feature film. This acclaimed commercial success was the culmination of Pixar's effort to improve their technical skills but, aside from that, the film continues a tradition of animated cartoon that goes back to early Disney's successes (Darley, 2002).

It is possible to say that computer animation has points of contact with established aesthetic conventions and forms from traditional animation (Darley, 2002). The kinds of objects, sets and props rendered and animated in an anthropomorphic manner are examples of these inherited conventions. Cartoon itself is an important artistic tradition that was – and still is – adopted and used vastly by computer animation.

Cartoon is a style of visual representation described by a simplification of the form to increase its communicated meaning. "Amplification through simplification" as described by (McCloud, 1993). The simpler form also has the characteristic of being less unique and broader in the sense of its representational quality. While a photorealistic image of a person depicts that exact individual, the cartoon of a face has a more "generic" look making it possible for different individuals to feel represented by it, as discussed previously.

Many characteristics of the cartoon aesthetics were transported to computer animation but also to digital games. Take, for instance, the staple of cartoons that is the use of anthropomorphic animals as characters. Early digital games used this element, even before the games had computational power to display a high-resolution figure on the screen. Arcade games like *Pac-man* (1980) and *Donkey Kong* (1981) had their protagonists illustrating the machines' cabinets, even if their 8-bit graphics would not suffice to depict their figures with the same level of detail.

Figure 15 - Original Arcade cabinets (Pietrak, n.d.).

Part of the game industry saw their products like toys, as was the case of Nintendo at the end of the 20th century (Sheff, 1999). They adopted cartoon elements in their games, as this style is often connected to children's imagery. Additionally, as digital games became more popular, many TV shows, films, and comics have started licensing their intellectual property (IP) to game studios. Many of those already had a cartoon style in their original media and had to be translated to digital games in a recognizable way. *Teenage Mutant Ninja Turtles* (1989) and *The Simpsons* (1991) are two examples of a practice that continued in every console generation.

Figure 16 - *Teenage Mutant Ninja Turtles*, 1989 (AL82 Retrogaming Longplays, 2014).

This cross-referencing from different pop culture media artifacts permeated the entire history of digital games development. From the very beginning, when games were based in form and content on "science fiction, Tolkienesque-fantasy and pinball" (Darley, 2002. p.25), until today, when independent games draw reference from those early games themselves. Whether being a direct source, such as in IP licensing, or as an influence, both digital and traditional forms of animation play a key role in game aesthetics (Darley, 2002).

2.3.3 The aesthetics of Independent Games

Digital gaming may have a brief history compared to other media, but it's certainly full of events. In its roughly 60 years, it went from a technological novelty to a commercially viable product and further to a worldwide multimillionaire industry.

From this commercial ecosystem of studios, publishers and retailers, a mainstream, or AAA, game development culture was formed. Described by Nadav Lipkin (2013) as "corporate in nature and capitalist in ethos" (p.9) and, despite not being unanimous, it can be considered as "emphasizing profit and popularity over creativity and artistry" (p.9). Juul (2019) points to the period between 1980 and 2005 as a time when the industry consolidated the conventions for how to develop, design, promote, and distribute games.

> "During this time, video games
> - Were involved activities on which players had to spend hours at a time;
> - Were sold in boxes;
> - Targeted males aged 10-35; and
> - Games (and specially hardware) were promoted on the basis of technically better graphics. Graphics did not have to be beautiful or pleasing but had to demonstrate the capabilities of the game console and technical skills of the developer" (Juul, 2019).

As in other industry, such as music (Hesmondhalgh, 1999 apud Fiadotau, 2018), a countermovement, unhappy with this scenario, emerged to go against the market practices they disagreed with. The independent game movement, or *indie games* as it is better known, is the name given to this movement that can have a hazy definition since it is not a genre in itself but a "a set of tendencies regarding how it is described by participants and outsiders" (Lipkin, 2013, p.10).

To better understand indie games, Juul (2019) classifies three types of independence – *financial*, *esthetical*, and *cultural* – and points to how they express the evolving concerns around indie games.

2.3.3.1 Financial independence

Financial independence came first, as it meant a clean break with the commercial ecosystem and their involvement in the game creation process. This made possible for the emergence of a new rhetoric on indie games being authentic (Juul, 2019), which means that digital games may seem representative of an author's expression rather than a profit-hungry AAA game.

This aspect has clear implications. The lack of funding makes these games have a much tighter budget, and their development will have to take this into account. From studios with dozens of employers, indie games are made by small teams focusing on small productions and controlled distribution. These economic limitations and the shared beliefs that define elements of the movement will, eventually, be responsible for the origin of a unique indie style to set them apart from the mainstream (Lipkin, 2013).

2.3.3.2 Esthetical independence

In esthetical terms, Juul (2019) describes an initial phase in which indie games still followed the visual style stipulated by the mainstream – basically, photorealistic 3D productions. This can be observed in the winners of the Independent Games Festival (IGF) from different years. Winners from 1999 to 2004 appear as “small versions of bigger-budget games” (p.58) that may still desire to use the award prestige as leverage for acquiring a publishing deal and be sold by mainstream standards.

After 2005, the winners seem to fall under the *Independent Style,* a categorization created by Juul (2019) that aims to encapsulate the visual representation strategies adopted by indie games at the time. The style is defined by the use of “contemporary technology to emulate low-tech and usually cheap graphical materials and visual styles, signaling that a game with this style is more immediate, authentic, and honest than are big-budget titles with high-end, three-dimensional graphics”(p .56).

Looking at three highly acclaimed indie games, such as *Cave Story* (2004), *Braid* (2008), and *And Yet It Moves* (2009), the appearance of each game has unique characteristics, but all three refer to an older, low-tech, visual style. *Cave Story* (2004) aims at the 16-bit era of Megadrive and Super Nintendo. *Braid* (2008) refers to traditional painting and illustration, and *And Yet It Moves* (2009) emulates an analog look, with an appearance that looks like it was made from paper cutouts.

Figure 17 - *Cave Story* (2004), *Braid* (2008) and, And *Yet It Moves* (2009), respectively.

These examples not only illustrate these visual style definitions but also point out another fundamental characteristic - a stylistic nostalgia born out of denial of the present (Lipkin, 2013). This nostalgia commonly refers to a historical moment linked to the developers’ youth - hence the huge amount of 2D pixel art games - but it can also suggest a non-existent historical moment. There was not a moment in time when digital games were made with paint over canvas or paper cutouts neatly organized. Even if the reference is counterfactual, these games aim to propose a simpler, earlier time (Juul, 2019).

2.3.3.3 Cultural independence

Indie games are often praised for reviving genres or giving a new twist to classic mechanics (Fiadotau, 2018, Juul, 2019), but after some time, neither this element, an original visual style, nor independence from the big companies was enough to consider a game indie. Games have to defend an ideology, to carry a cultural, political, and moral message (Juul, 2019). *Braid* (2008) ultimately can be read as a critique of sexism and male chauvinism, but later games, such as *That Dragon, Cancer* (2016), have a much more explicit *message* to convey.

In an implicit search for its position in the community, this phase of indie games saw the earlier implementations too close to mainstream titles as those games were merely fun. Meanwhile, voices from the same community sometimes declared these new games as exaggerated or "inauthentically trying too hard" (Juul, 2019, p.25). What it means to be indie continues to change as these three defining facets interact.

2.3.4 Contemporary Indie Games Scenario

In 2005, the independent style marked its presence in the indie game scenario. However, it also marks the increasing digital game distribution, with *Valve* beginning to negotiate with several third-party publishers to release their products on their digital platform, *Steam* (Strategy First, 2005). Indies not only did commercially well but also created a community of developers, players, critics and specialized news channels, such as the *Indie Game Magazine.* Indie gaming's success and popularity brought the attention of mainstream companies that saw a new market to invest. The independent style started to be emulated as it began to be co-opted by the mainstream (Lipkin, 2013).

Not only the style but the term itself is co-opted, creating a "semantic confusion" (Lipkin, 2013, p.7). Its meaning loses clarity, creating confusion in differentiating what an indie game is from what it is not. *Minecraft* (2009) is a game created under the indie circumscriptions as it was: developed by one person, Markus "Notch" Persson; it had a publisher, *Mojang Studios,* but the starting screen shows the message "Made by Notch" to indicate the personal nature of the project (Lipkin, 2013); its appearance is a blocky 3D reassembling volumetric pixels (voxels) in reference to the simplicity of pixels itself; and it was acclaimed by its emergent gameplay mechanics. Nevertheless, the game was bought by *Microsoft* in 2014. Can it still be considered an indie game?

Figure 18 - A scene from *Kena: Bridge of Spirits*, 2021 (Kimber, 2021).

A more recent case comes from the 2021 edition of The Game Awards (TGA). The game *Kena: Bridge of Spirits* (2021) won the *Best Indie Game* category. It is the first game developed by a small studio called *Ember L*ab, which, in turn, was backed by *Sony* to launch the game on their *PlayStation 5* console (Schreier, 2021).

Putting aside the doubt about the meaning of this prize for the indie community - a victory of a small studio or the success of a mainstream publisher in co-opting an independent space – the game's appearance was a leading factor in its success. Its visual style was explored as an important selling point (Kimber, 2021), a strategy associated with AAA games. However, rather than having the staple photorealistic look, the game is often described as having a Pixar films style. This matter signals how blurry the lines between mainstream and independent can have in a complex market scenario as we have today.

A confusing scenario, as exemplified above, is where indie – and mainstream games – find themselves at present. As concluded by Lipkin (2013), even if the scenario is confusing, it is noteworthy that the AAA companies are reaching out to small studios to create "smarter, more innovative, and more personal titles that enrich the community, rather than merely providing distractions to them" (p.14).

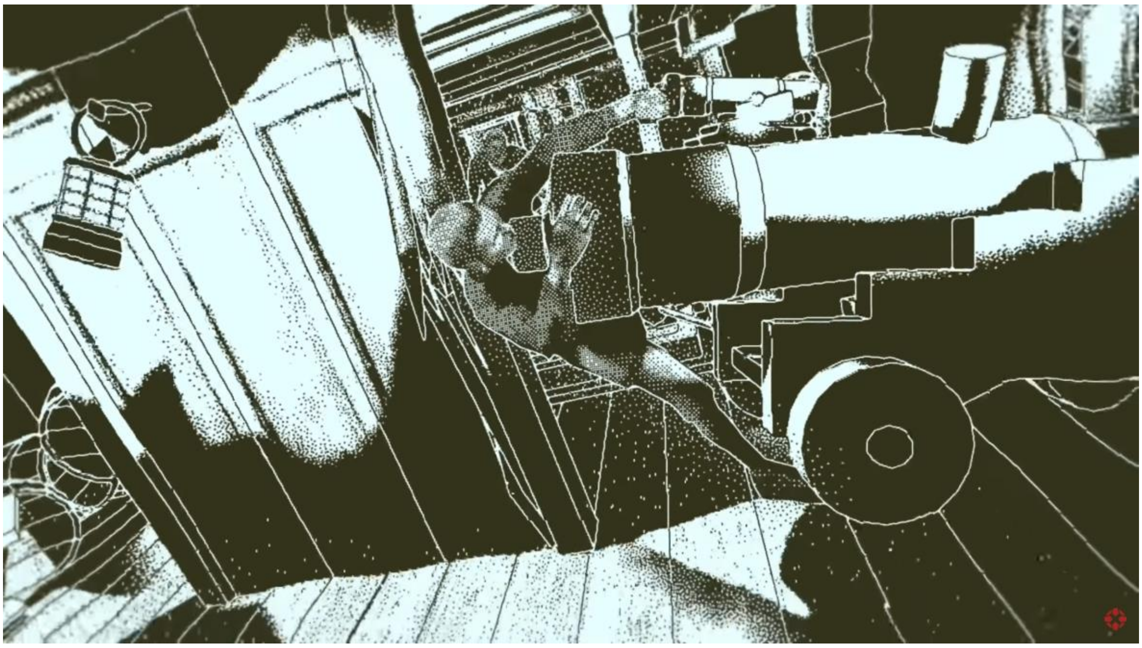

Figure 19 - ***Return of Obra Dinn*****, 2018 (Marks, 2018) is a 3D game with a dithering treatment that makes it resemble 1-bit pixel graphics. Two extremes brought together in a uniform appearance.**

In terms of visual style, the once clear division became indistinct as AAA games adopted some elements explored in indie games, and independent developers explored the photorealistic style as its entry barrier lowered. Game appearance remains an important aspect as, on the one hand, it is a sales argument used by the mainstream publisher to leverage their sales. On the other, it forms part of the aesthetic expression designed by the game creators and will continue to be explored in new and exciting ways.

This chapter began by describing the defining elements that constitute visual style in digital games, introducing fundamental definitions and terminology to understand the dissertation's framing and field of action. Following up with the photorealistic and stylized styles survey, shedding light on the meaning of these common use terms and bringing up matters of interest in both styles. More than ending an old discussion on the subject, this research aims to make way for more exploration of the topics at hand, especially considering the current scenario of the gaming market. It seems timely to study the impact of each visual style on the aesthetic experience of digital gamers.

3. Project Development

3.1 Intro

Using the information gathered in the previous chapters as a theoretical foundation, the research advances towards developing a prototype in which the concepts raised can be applied practically. This will be a prototype of a digital game that uses both stylized and photorealistic visuals and will serve to describe and understand the technical-artistic challenges of creating and implementing each of these styles in a solo development scenario as defined as the main objective of this dissertation.

This stage started with a planning phase to define the development methods for the prototype creation, its structure, and parameters to consider as the project advanced. An important aspect to consider was that a single person would lead the whole creation process. As discussed before, developing a game is a complex task and in this scenario of a solo developer, one has to assume many hats. Pluralsight (2015) lists seventeen different occupations that are usually found in the game industry. Three of which are related to matters of design; eight, to art related subjects; and another three are related to programming and implementation. The others relate more to concerns outside the scope of this dissertation, such as Network Administration or Quality Assurance.

In a broad overview, the design occupations listed are Narrative Designer, Game Designer and Level Designer. The Narrative Designer is responsible for the story creation and how it will be delivered within the game experience. The Game Designer is responsible for the game aspects, rules, and challenges of the experience, and the Level Designer will plan how the environment will unfold for the players to afford the rules of the game and its story in a coherent manner.

The art-related occupations listed in Pluralsight (2015) are Concept Artist, Environment Artist, Character Artist, Technical Artist, Animator, Art Director, UI Artist, and FX Artist. The concept artist creates images to illustrate the ideas for all game-related matters. Their characters, locations, objects, collectables, and any other element will appear on the screen. After that, the character and environment artists use those concept images to create the assets that will be used in the game itself. In a complementary job, the animator is responsible for animating the characters and all the other elements that will display motion.

UI (User Interface) artists are responsible for the creation of UI elements such as menus and HUD elements. FX (Visual Effects) artist makes effects that usually require simulation of particles, such as explosions, water flow, or fire effects. At the same time, the technical artist ensures that the assets made by the previous artists work in the game engine creating tools and scripts to help with this integration process. Lastly, the art director is responsible for coordinating all those tasks, so they all work together coherently.

Regarding programming, Pluralsight (2015) lists Game Programmers, AI (Artificial Intelligence) Programmers and Pipeline Tools Programmers as common occupations in the game industry. The game programmer is responsible for taking assets and designs to work in the game engine. AI Programmers focus on the artificial intelligence elements, such as enemies and other non-player characters' behavior, while Pipeline Tools Programmers will work to create tools to help the work of all other departments and ensure that the pipeline of developments flows well.

Back to the single developer scenario, not all the skills covered by these professions will be used because the more specific tasks arise according to the size and complexity of the game. AAA studios have a large team of expert professionals but aim to deliver long games with multiple scenarios and diverse mechanics. This factor can already be considered when defining the type of game to be developed at this stage. Knowing its limitations is fundamental for the definition of the project.

3.1.1 The Project Triangle

Therefore, as a solo designer, it is also important to recognize your weaknesses and limitations to understand how far you can truly develop on your own and in which aspects you will need support. This awareness has an impact on the way the project develops. In addition to personal limitations, others may arise during the development of the prototype. To help systematize these limitations, the project made use of the project triangle: scope, cost, and time, considered as parameters that guide these limitations.

Project Development

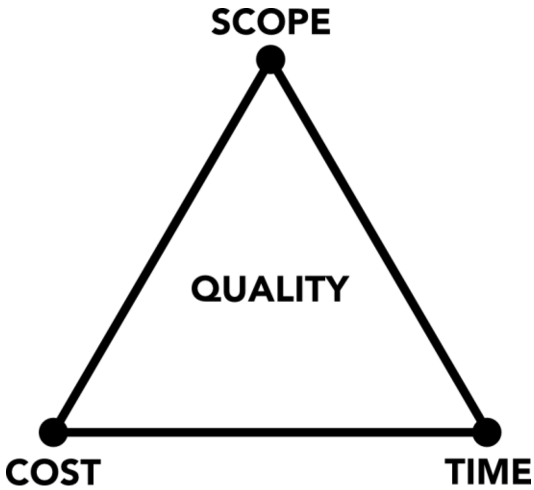

Figure 20 - the project triangle diagram

In the project triangle each vertex corresponds to a design variable; therefore, as in the geometry, whenever one of these edges is modified, there will be a need to compensate for this change at least in one of the two other vertex. The balance between the base parameters is the responsibility of the project manager, who aims to meet the project specifications within a given time and within budget (Asana, 2021). Asana (2021) concludes by saying that the lack of these adjustments can cause the triangle to break; that is, the quality of the design will be impaired.

The development of this game prototype can be seen through the lens of project development, and as such, it is possible to apply the project triangle to it. In this regard, the Time variable is defined by the deadline for submission of the dissertation itself. It must be possible to be executed within this period with time off for contingencies and for the subsequent writing of this report, which in this case was for about two months. The Scope is the definition of what the prototype should present at the end of the available time. The minimum expected to fulfill the objectives of the dissertation would be the creation of a game prototype that uses photorealism and stylized visuals, but this is too little information and need to have a deeper definition so the prototype can be defined.

As this is an academic work, there is no defined budget; however, since this project also aims to simulate the conditions of game development by a single designer, the ideal would be to spend as little as possible. However, since the deadline is short and fixed and the scope must comply with the dissertation requirements, the budget aspect is variable with more flexibility. This implies that, if necessary for the completion of the project, the acquisition of assets will be an option to be considered. These assets can be anything that cuts repetitive work or work outside the main concern of the project, that is, with the visual outcome of the game. 3D asset packs, texture packs, tools and third-party programming services are some examples of acquisitions to expedite the project. However, the analysis of the real need for each purchase will take place during the project, if necessary.

3.1.2 Design Methodology

Because the prototype development is a project in and of itself, the question of how to approach it emerges as a critical one that must be addressed before its execution can begin. There are various ways to approach a project, especially since diverse industries deal with project development. As a result of the various scenarios, a variety of methods for unraveling a project from start to end have emerged. It is up to its developers to choose a method that suits their needs.

The Design Sprint Methodology (Google, n.d.) was chosen as an option to guide this project. This methodology was developed by Google to be a method to evaluate ideas by small teams in a short period of time. It was created for sprints, which are five-day sessions in which a product idea is explored until it reaches the level of a prototype. Even though the time frame is completely different from the two months that were available, the concept of transforming an idea in a working prototype fitted well with the goal of the project.

The Design Sprint is organized into phases that allow you to focus on the components of the project that make the most sense at that moment in time. These phases are as follows: *Understand*, *Define*, *Sketch*, *Decide*, *Prototype*, and *Validate*. They not only serve the development of the project itself but will also work to organize the sections in this chapter.

The final element that worked together with the project triangle and the Design Sprint to shape and define this project was this author's individual experiences. This tacit experience came in the form of assumptions for the project's development, based on more than ten years of experience with design-related projects but a lack of understanding of appropriate game design and development. These assumptions can also work as reference points in the end phase of the project as they can be revisited and compared with the reality of the execution and can lead to new and useful insights.

The following is a list of these assumptions:

- The development of the photorealistic appearance would be, in general, harder than the stylized one;
- The work of 3D modeling would be the most arduous and would require more execution time, which could become a bottleneck for the project development;
- Creating an interesting 3D environment would be a straightforward, easy to achieve, process;
- Great support would come from the Unity community on the internet, via blogs, vlogs and forums. Any problem found during development was already solved, and its solution was made available online.

Now that the project's basic framework has been established, it is time to begin its development, beginning with an understanding of the problem to be solved.

3.2 Understand and Define

In a teamwork setting, it is critical to establish a shared knowledge base among all participants so that all variables and points of view can be discussed and argued, resulting in a unified picture of the project within the team (Google, n.d.). This matter is not accounted for in a single-person project, so the main concern in this phase was to be aware of the task and to define it as best as possible to ensure a smooth prototype development.

This stage of the project relates to the game designer's early work as it is their task to define and share their established vision for what the game should be. Conceptualizing what experience the game will deliver for its users and creating goals and progression while focusing on system consistency is primarily what game designers do at the start and throughout game development.

What had to be accounted for in the instance of this project was a sequence of intentions to construct the game's initial idea. The fundamental intentions were, first, to create a three-dimensional explorable environment. Secondly, that this explorable environment could be displayed with a photorealistic appearance and a stylized one. Its elements would be the same, and only the visual style would change. Lastly, that the game prototype could be narrative genre agnostic i.e., that the game would not fall easily into one of the classic categories such as terror, fantasy or sci-fi. To fulfill this intention, the environment created should be fundamentally generic/familiar in the digital games community so that it would not be linked to a narrative genre.

Another goal was to create simple game mechanics such that the time it took to implement them would not significantly impact the overall development duration. Suppose the aimed level of complexity was too high. In that case, most of the two months could be spent on its implementation, rather than researching and constructing the planned environment and visual appearances, due to a lack of prior expertise with the essential programming to create these mechanics.

These intentions, together with the project triangle points of concern, guided the necessary definitions of the project. These definitions are now presented grouped by category.

3.2.1 Environment definitions

For the environment, three areas were chosen to be depicted for their common presence in various games. The first things that came to mind were forests, ruins, and city blocks, but after some consideration of the amount of work required to create the three areas, the city block was reduced to a single urban element in the form of a gas station and a road that could be integrated more cohesively into the overall environment.

The ruins also had to be adapted as the term alone could be descriptive of many types of possible ruins, so it ended up becoming an abandoned military base area in its place. The forest was the only set that was maintained from the start. The usage of forest environments in games is enough commonplace to avoid any compromises with the project's goals.

A cabin was added inside the forest to be a landmark and the starting point of the game. Cabins in forests are also staples often used in games with various genres, and the decision to utilize it would not impact that premise.

3.2.2 Production Pipeline

Another important definition had to do with how this project would be, in practical terms, accomplished. Choosing which technology would host the digital game would have an impact on the project as a whole since it could impact any of the vertex of the project triangle.

The main technology definition had to be the game engine. The engine is responsible for expediting most of the digital game creation as many of the abstractions a game may have — such as basic physics system or light behavior — are already implemented and presented in a single solution. This way, game developers can focus on the elaboration of the particularities of their game rather than having to invest time in fundamental logic.

From the many options in the market today, *Unity* has been the chosen game engine since the beginning of the project. It has many qualities, such as being a professional tool used by many game studios and present in many commercial successes. It also has a Unity Free/Personal license that allows the use of all its functionalities and even allows the selling of commercial games, a critical use case for the solo-dev scenario of this project. However, above all that, this was already a familiar game engine for this author, so its entry barrier would be more easily overcome. Also, the engine has a strong community around it that not only shares knowledge but also developed a market of tools and a variety of game development elements for Unity use.

With the game engine definition also came the definition of C# as the programming language for the project since this is Unity's adopted language. Microsoft Visual Studio 2019 would be the IDE for writing the needed scripts.

Other important software chosen for this project was related to the creation of the 3D assets. The choice was also based on the principle of familiarity with the systems. *Blender* was the choice for 3D modelling software. Beyond the authors' familiarity, Blender is free and open-source and has, over the years, gained increased space within the game development community, today being a frequent choice by independent studios. By the same principle, *Substance Painter* was chosen as texture creation software for the project. This is not free software, but one already owned by this author. This software is also a common choice in the development of 3D assets for games and would serve well in this scenario.

A concern that surfaced early on was the game's ability to support two distinct visual styles. The defined environment was complex enough to be created in one visual style, since the work of modeling and texturing a 3D asset is very time-consuming. To model and texture several assets twice would take more time than was available, so a decision had to be made on how to include both styles. The method devised was to prioritize the creation of assets for the photorealistic scene and reuse them for the stylized, altering only its shading characteristics.

In terms of CGI, a shader is an algorithm which mathematically describes how an individual material is rendered to an object and how light interacts with its overall appearance (Peddie, 2013). Unity has a whole section dedicated to shading and even a proper coding language to deal with this topic. The last addition to Unity's toolset of shader-related systems was *Shader Graph*, a tool that allows the creation of custom shaders in a visual way instead of writing code (Unity Technologies, 2022). The use of Shader Graph would allow the creation of all the shaders necessary for the stylized scene.

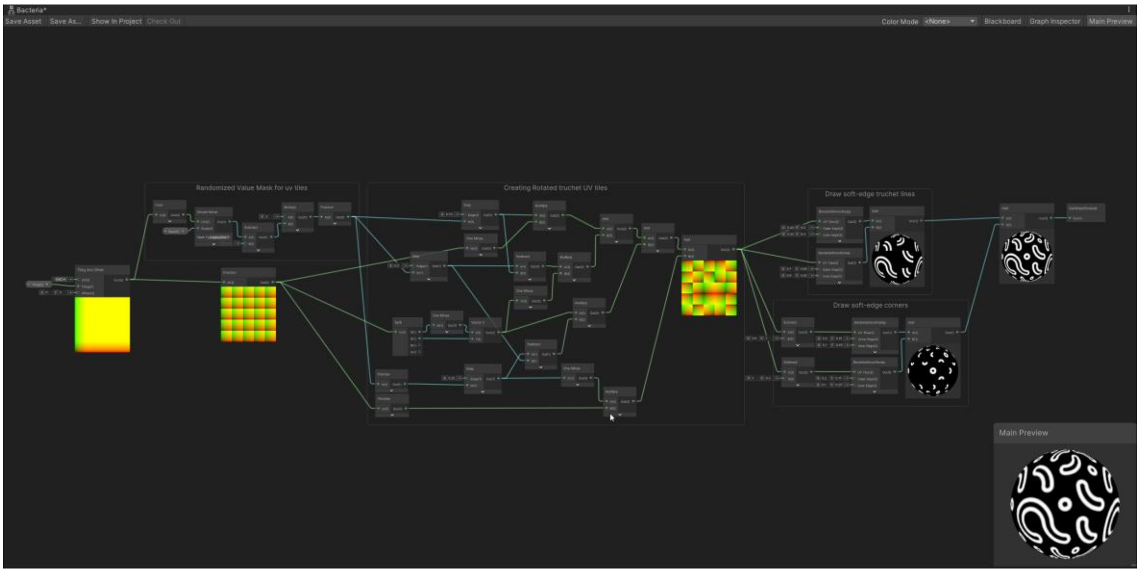

Figure 21- 'Bacteria' shader from Shader Graph examples

The choice to privilege the photorealistic scene and the use of the Shader Graph tool defined the Render Pipeline to be used in the project. Unity, today, has three rendering pipelines, i.e., the mathematical process to calculate light behavior and, ultimately, the game's image qualities (Unity Technologies, 2022). The High-Definition Render Pipeline (HDRP) was the one that had access to the Shader Graph and was also built to deliver photorealistic results, among the options.

This compromise may have an impact on the game's aesthetics. Because stylized assets are derived from photorealistic ones, they may feel uncanny. The results could be less satisfactory than if they were created from the start. The evaluation of this decision – and many others – would be possible at a later stage and, for the time being, would suffice as it restored the project's viability.

3.2.3 Game Mechanics

As stated above, the intention was to create a set of simple game mechanics for several reasons. Firstly, the objective of the project was to focus on the 3D environment, and it would not make sense to spend much of the development time on the mechanics and leave the environment aside. Secondly, our experience has more to do with 3D asset creation than with programming, so to create an elaborate game mechanic could prove to be an impossible challenge that could compromise the whole project.

As the creation of a navigable space was one of the main intentions, some sort of spatial movement mechanic had to be implemented so the player could explore the scenario. The matter of point of perspective was also relevant at this moment, and it would impact the experience of exploration. A first-person perspective was chosen as the best fit for the project as it provides an unmediated experience. With these decisions, the first idea was to create a simple walking simulator in which the player could only experience the environment without any other game element. After some discussion, it was agreed that the project needed more game elements to be completed and *fun* to be achieved.

Initially, a mechanic of collecting items was idealized as a viable option for the prototype. The items would be spread in the space, and the player had to collect all to advance to the next level. This created a possibility to link the photorealistic scene with the stylized one. But a problem arose from this decision: how to let the player know what they were supposed to do?

To solve this matter, another game mechanic was introduced. A player monologue in the form of text at the bottom of the screen would appear in pre-defined moments and locations so the player could receive hints of how to move forward in the game. This mechanic introduces narrative elements that were previously avoided in order to avoid falling under a narrative genre. To mitigate this matter, the idea was to create vague and undescriptive text for these monologue moments to avoid explicitly choosing a genre or known scenario that could lead the player to a specific genre.

The tandem between 3D space movement, item collection and narrative encapsulate all of the planned game mechanics and had the promise to lead to a proper game experience that would satisfy the project objectives and its players.

To summarize, at the end of this phase, the defined points were as follows:

- Game Engine: Unity
- Render Pipeline: High-Definition Render Pipeline (HDRP)
- Dimension: three-dimensional space
- Point of perspective: first-person perspective
- Visual Outlook: photorealistic in one level and stylized in another
- Soundscape: out of the scope of the project
- Senso-motorism: the most straightforward approach would be developing a PC game, so keyboard and mouse were the main input mechanisms for the game.
- Core Gameplay Mechanics: spatial movement, item collection and monologue system.

3.3 Sketch and Decide

After the project scope was better characterized by the definitions made in the last phase, it was time to generate a broad range of ideas (Google, n.d.). In a digital game studio, the Concept

Artist would perform this role. Their responsibility is to help teams solve problems by designing visual solutions (Riot Games, 2018a).

This task started with research for inspiration from related works and other media. A reference board was created from images found on different websites and digital games. After a first selection of the most relevant ones, the software *PureRef* was used to organize the images grouped by theme: cabin, forest, road, gas station, and military base.

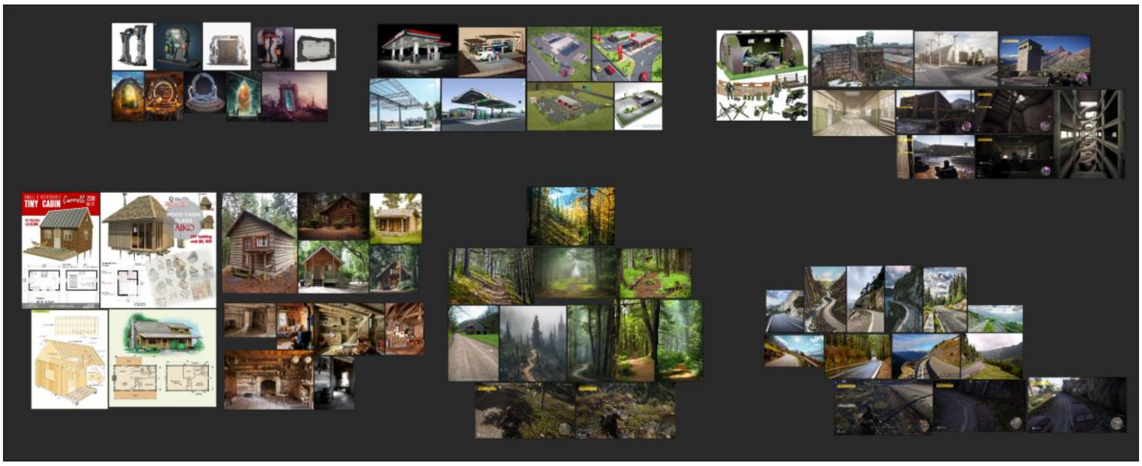

Figure 22 - Reference board created in PureRef

The chosen images can be grouped into two categories. The first are photos of the different elements. In particular for the photorealistic style, these provide direct pointers to how the elements appear. The second are game screenshots, useful as they depict the element as portrayed in the final media and can be a source of information on the technicalities of the creation of said element.

Following that, sketches of various aspects of the environment were created to begin transforming some of the ideas into a more concrete vision of what it would look like. This sketching phase uses fast and loose drawings because the goal is not well-finished illustrations but the materialization of concepts for the game.

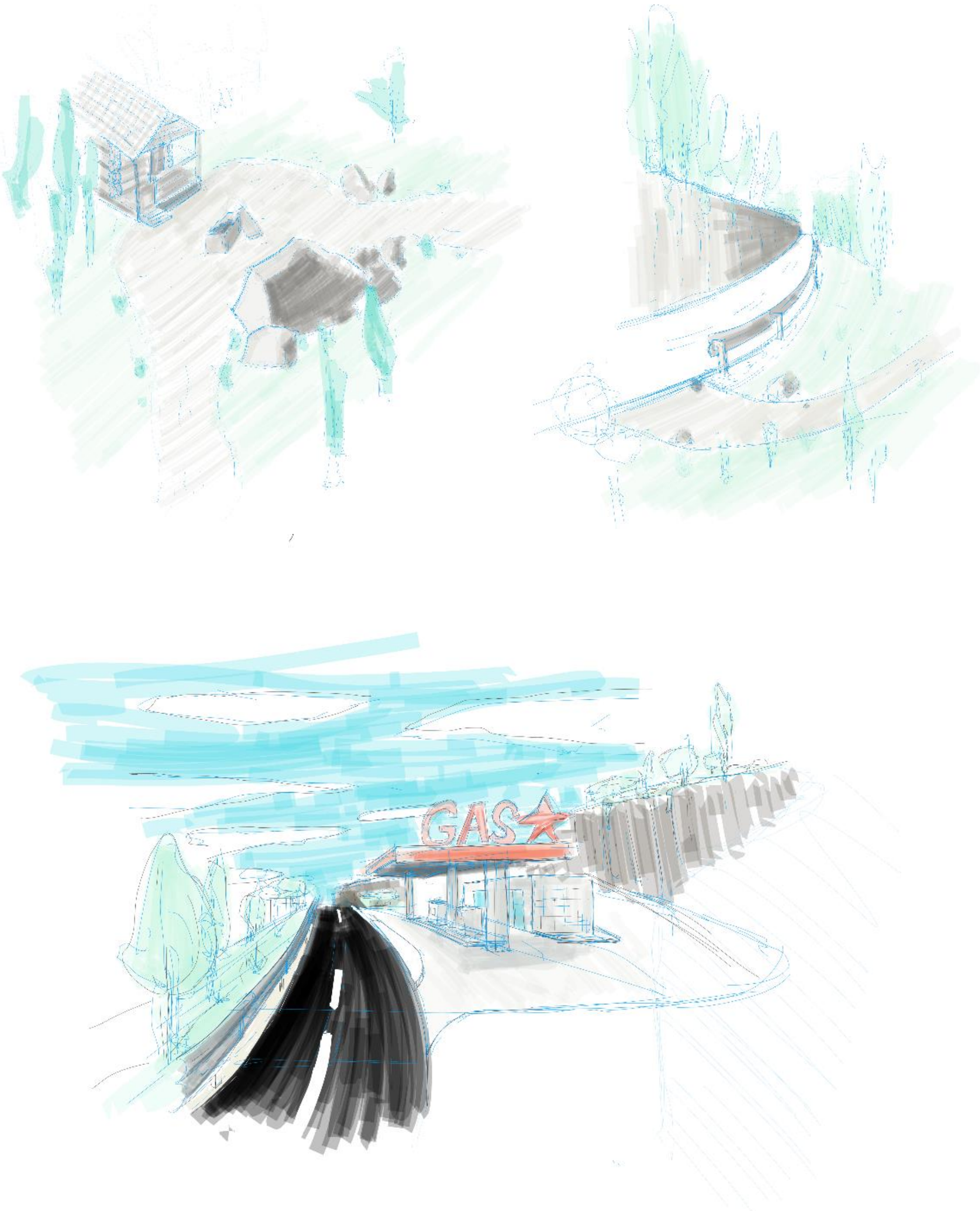

Figure 23 - sketches of some of the main areas of the game

In parallel to the concept creation, a map of the level was created so it could be implemented in Unity later. This task would fall under the design team's responsibility, specifically for the level designer. In this stage, their work would be to create a map considering internal cohesiveness, balancing environmental challenges, and promoting an understandable navigation for the player.

A schematic drawing started this process as the first concern was to relate the scenarios between themselves.

Figure 24 - schematic map of the environment

From the start, it was decided that the forest set would be the element that would connect all others together. At this stage, three support environments would be created to be used as non-navigable, eliminating the need for the infamous invisible wall to restrain the players' movement, but this decision was reversed because it would allow for more work than the time constraint would allow. This schematic was a great tool as it helped a lot in making this kind of ruling to trim the level to what was possible to be made.

From the schematic to the final map, changes would appear as decisions from the game design and level design sides would have to be accounted for. The city blocks were replaced by the gas station area and a road; the three non-navigable environments were reduced to a single one, namely a Mountain Range to surround all the level; more cohesion was added as the map would depict a portion of a road and its surroundings.

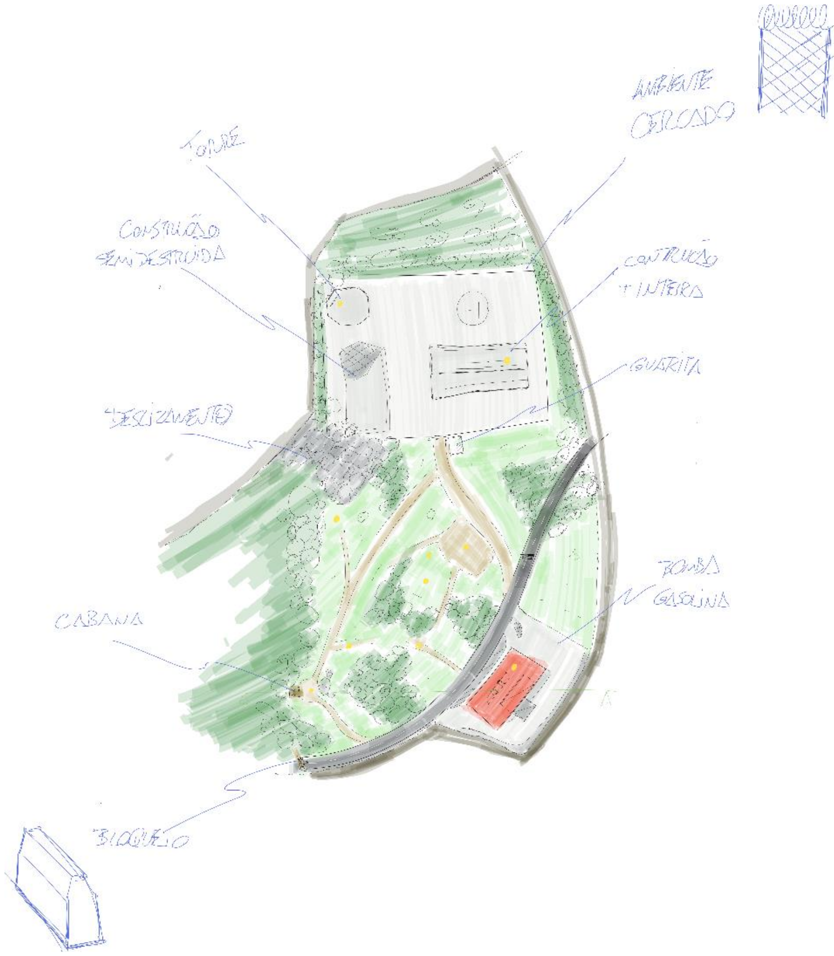

Figure 25 - Final map with annotations

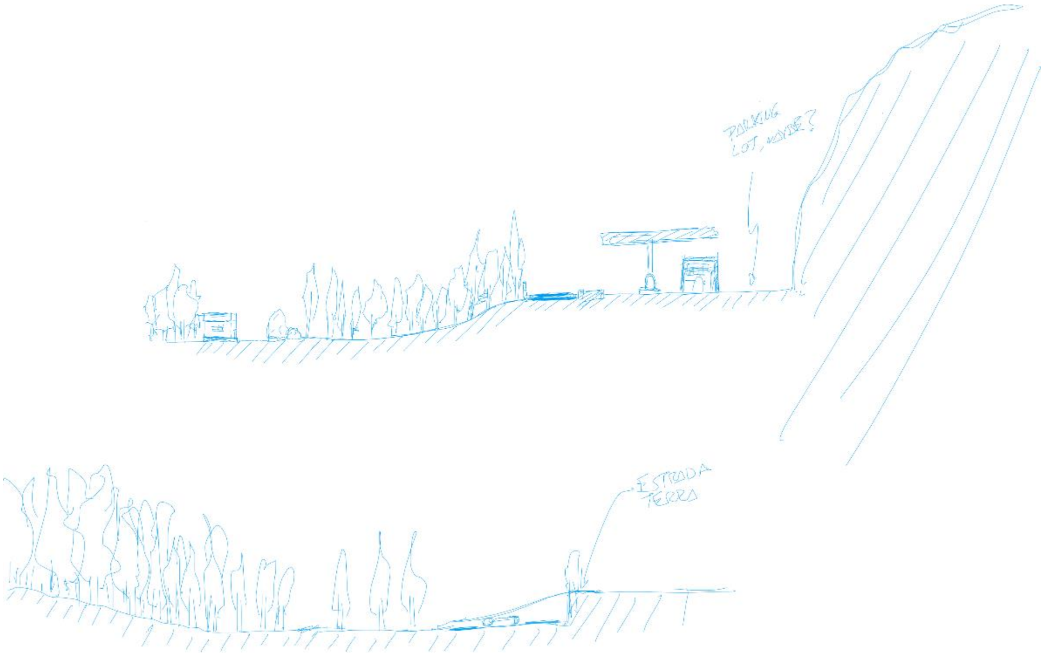

Figure 26 - A introductory study of the topography for the prototype environment

Another level design concern was the size of the map. Defining the size was important because it is one of the aspects that defines the time a player spends on each level. It is easy to understand that a smaller level would need less time to be explored, whereas a bigger one would take more time. Using the movement speed as a base, the size has been calculated so that it takes players approximately 5 minutes to walk from the cabin to the end of the military base. Considering 5km/hour as the base walking speed, this distance would be 400m.

With a map concluded and some concept drawings of main locations done, Design Sprint had finalized the direction to be prototyped. It was time to implement these decisions in the game engine.

3.4 Prototype

Design Sprint projects aim to create a prototype that is just real enough to be valid (Google, n.d.) and are built to be prototyped fast. This was the case with this project, but because of its nature – being a game designed by a single person – developing the prototype itself took the majority of the two months set for the project.

Its development started in Unity version 2020.3.36f, the latest LTS (Long Term Support) version available at the time. Working whit LTS is always preferable as these are assessed versions with more support and less prone to crash or to have bugs.

Unity works with a system in which each level is constructed inside a *scene*. As mentioned before, the first level to be implemented would be photorealistic, therefore a scene was created to contain all photorealistic environments.

3.4.1 Photorealistic Level

3.4.1.1 Terrain Construction

The process of construction of the terrain relates to tasks from both Level Designer and Environment Artist. While the terrain had to be designed to ensure that the level design decisions were followed, the topography had to be interesting and consider the buildings and constructions that would be added later.

The terrain was built with Unity's *Terrain Tool*, which allows the creation of land planes inside the game engine. The tool includes several brushes that allow for the editing of the height and shape of a terrain plane by controlling the mouse cursor in a manner similar to that of illustration software such as Adobe Photoshop. The difference is that in illustration software, the mouse cursor would add a color or a drawing line, and in Terrain Tool the cursor changes the heigh of the terrain plane and, later, also adds texture information to it.

The initial process was what is often called *block out* or *greyboxing*. In this process, the terrain topology is created with the help of simple cubes that work as placeholders for the buildings and other defining elements of the environment. In this phase, the main concern is with the design of the environment, its proportions, relations, and cohesiveness, so all elements remain with the default white/greyish color – hence the name, greyboxing. To begin, the map and other reference photos were used as a guide.

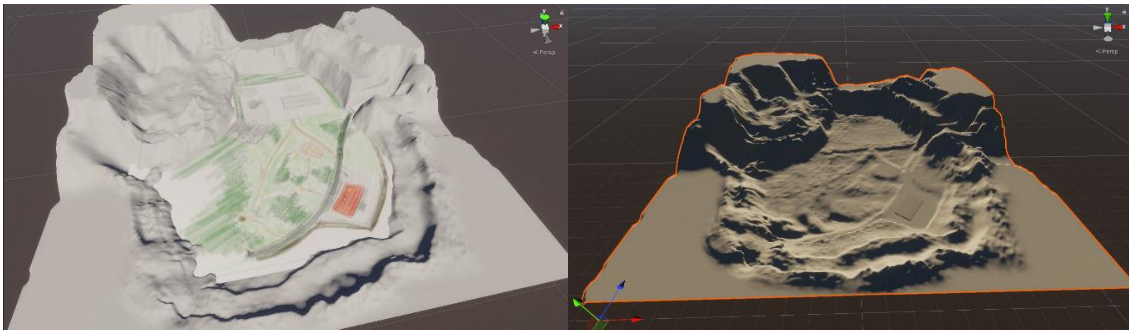

Figure 27 -The designed map serves as a base for the creation of the prototype terrain.

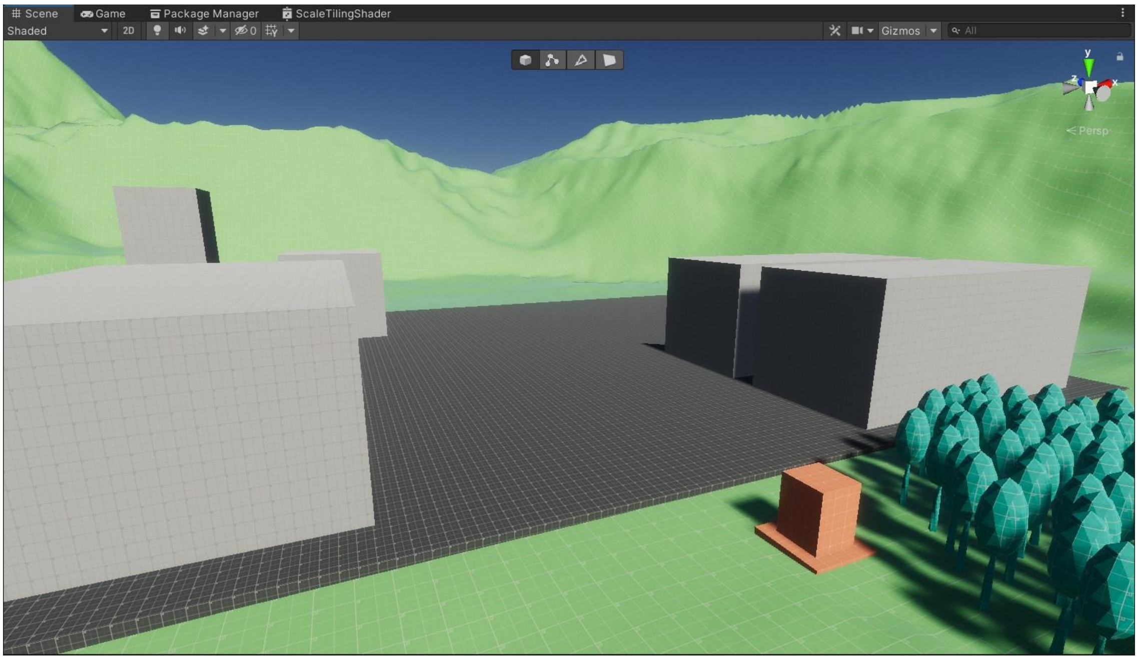

Figure 28 - Block out of the military base area. Here color was added with some placeholder textures to help highlight each element.

It is critical to pay attention to the size of the cubes because they represent a future game element and must be the same size as these elements. In the case of the gas station, research online returned that an average area for a set of gas pumps, maneuvering area, park space, and store could be around 1.500 m^2, an area ten times smaller than what was created at first. This information alone was sufficient to cause the terrain creation process to begin overdue the necessary adjustments in the surrounding area. This example can illustrate the fact that greyboxing is an iterative process. It is easier to make big topographical changes in an initial phase than at the end, where much more information is added to the scene.

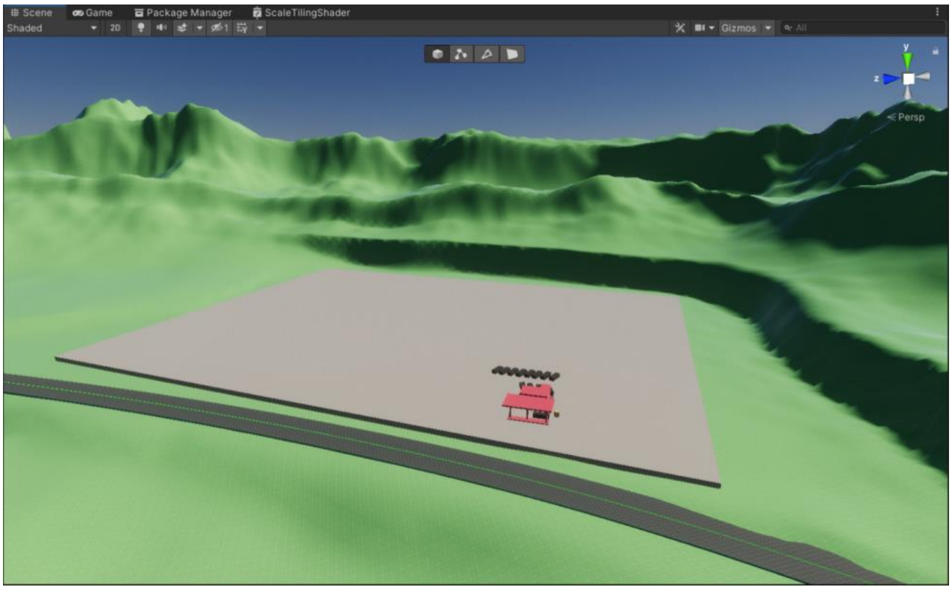

Figure 29 - All gas station elements compared to the area intended for them.

As this procedure continued, the use of the simple cubes felt limiting, so low polygon count (low poly) models were made to be used as in the main game areas. This substitution aided in the vision of the eventual result now that the terrain had been altered by the displacement of large land masses.

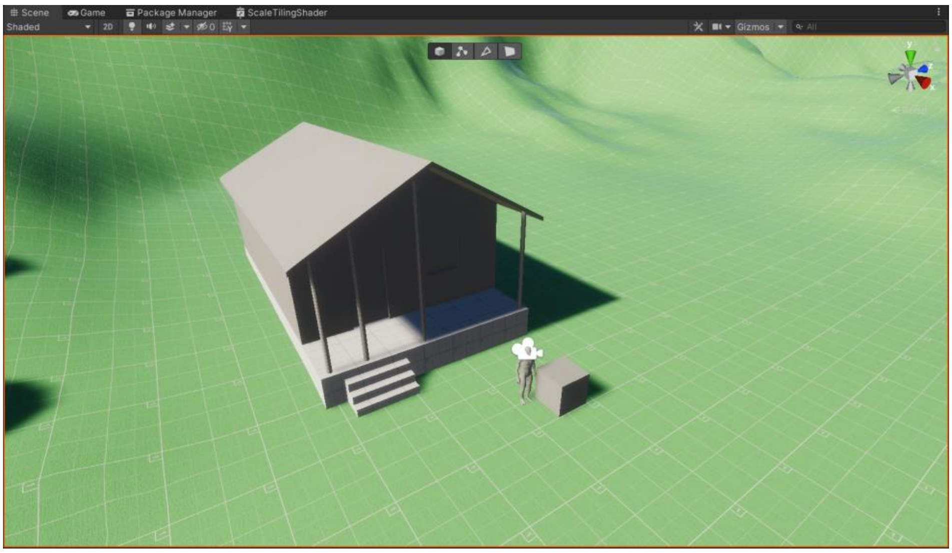

Figure 30 - The cabin was the first box to be replaced by a low poly asset. The human and the cube models were used to confirm the model scale.

The road was another landmark that would add weight to the level. Because its design would have an impact on the terrain itself, it needed to be editable so that both could adapt to one another. The way found to accomplish that was to generate the road procedurally by using a curve to guide its mesh. After some research in the unity community, a tool built by Sebastian Lague, called *Bezier Path Creator* and available in Unity's Asset Store, turned out to be a satisfactory solution.

It allowed the creation of a cubic mesh, guided by a 3d curve. In turn, this curve is driven by control points that adjust its shape, as shown below:

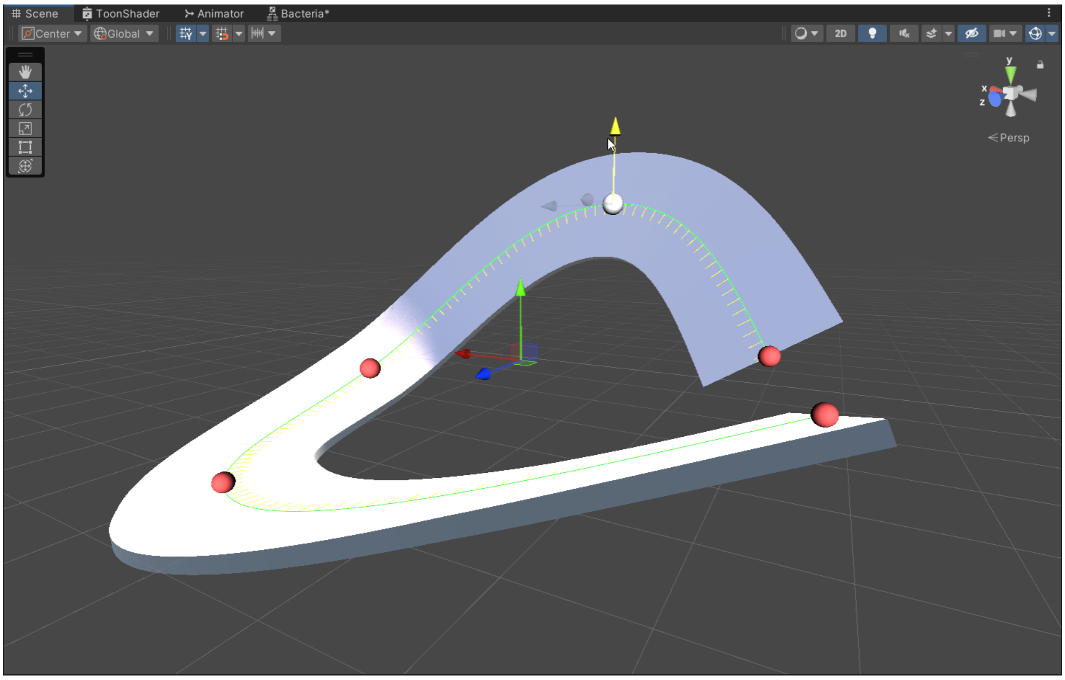

Figure 31 - 3D Curve controlling the shape of a mesh using Bezier Path Creator.

This tool worked well enough, but it only worked with a parallelepipedal mesh was inherently restrictive, in contrast to other paid tools that allowed control of more complex models. The drawback of sticking with it was that the road ended up being only the paving without the guard rail planned before.

A set of two placeholder textures were introduced to the terrain to assist in defining the trails that appear in the forest area. These textures came in a pack called *Minimalist Scalable Grid Prototype Materials*, also available at the Asset Store. This pack ended up being used throughout the scene as things got more complex and color would help define silhouettes and in the hierarchy of some visual elements.

The next phase would be to start to define the textures for the terrain. This is a tough task because the textures have to blend in a convincing way to be perceived as photorealistic. This process starts with the acquisition of quality textures that can be found in paid or free sources. The paid versions often come in a pack of several textures bundled together and can be found in many web stores, such as Unity Asset Store or even specialized stores like *Poliigon* that work exclusively with textures and 3d models. Many websites offer free versions, but quality textures can be difficult to find if it were not for initiatives like Polyheaven and AmbientCG, which are public asset libraries that offer textures and 3D models for any purpose without use restrictions. These libraries are a fitting example of the community that exists around independent game development.

After some iteration with a larger set of textures gathered on the above-mentioned websites, eight textures are chosen to compose the terrain. HDRP uses a group of three kinds of texture to each material as a basic setup. The first is called *base color* - or *albedo* - and, as the name implies, is responsible for color information; the second is the *normal* texture, which controls superficial height information; and the last is a mask texture that combines smoothness, ambient occlusion, metallic values, and another detail mask in its red, blue, green and alpha channels. This kind of solution is often used in games as it is less resource-intensive than having separated image files for each texture.

Figure 32 - Texture stack for the photorealistic level terrain.

The terrain tool is once more the means to apply these textures to the ground geometry. The brush-like method is used once more, but this time with control over brush shape, size and opacity. This is what allows the blending of the textures as they encounter each other on the same surface. Due to the overall level of experience with the tools, this procedure took a long time to complete. To get a meaningful result, it makes sense to require some amount of artisanship. In this scenario, practice makes a difference in the end product's quality.

At this point, it was noticed that the terrain ended up being without major slopes or inclines. From a level design point of view, the lack of height information may lead to a confusing space. Landmarks that could be visible from a distance were introduced as a means to mitigate this situation. A communication tower was added next to the military base, and a neon sign at the top of the gas station roof structure. To the cabin was added a particle effect to simulate smoke coming out of its chimney as this solution would not add a new asset to be created.

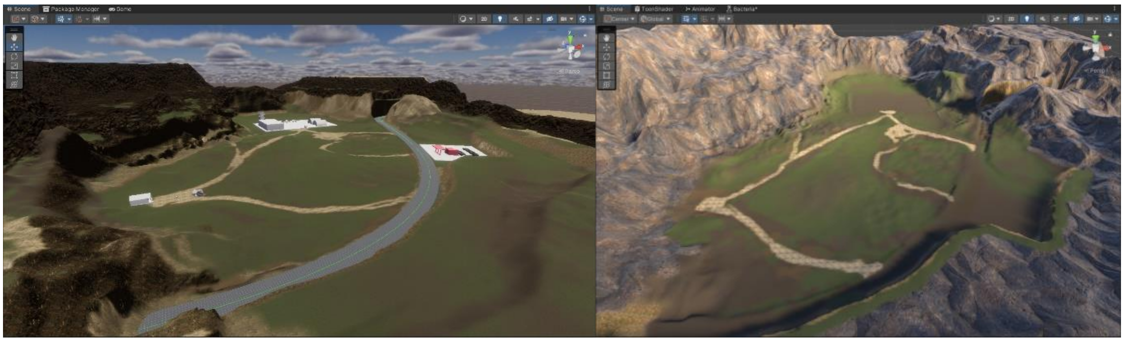

Figure 33 - The terrain creation process.

The terrain would not be completely finalized, as the final assets had not yet been added to the scene. Some final additions would still be necessary after they were introduced. For now, there was no point in continuing working with the terrain, even though it was still possible to continue giving it several tweaks and adjustments. The best course of action at this point would be to start the development of the 3d assets.

3.4.1.2 Assets 3D

The design of the 3d assets is the process that starts with the creation of the models, goes through texturing these models and the adaptation of these elements to Unity's environment. The game industry occupations in charge of this task are the character artist, if the asset to me created is one of the game characters, and the environment artist for the other elements. The Prop Artist position is also commonly adopted by the industry in studios with a large staff and more specific job positions. The prop artist is usually responsible for the creation of 3d objects — or *props*. Props can be designed to be close to the camera and the player — in which case it is referred to as a Hero Prop — or to be far from the camera solely to aid in the composition of the scenario.

Asset production for this project can be divided into three divisions based on the varied ways to complete a finished result. The first group would be the assets that were treated as hero props and had to follow a specific framework of development. The second group will be called environmental assets. This group includes buildings and larger objects and follows a different framework. And lastly, the group of asset packs that were bought. The necessity to buy assets comes from the project's time constraint but is a widespread practice in solo development scenarios as one person often does not have the skills to complete all tasks necessary to complete a game.

What was initially planned to be bought was a set of forest environment assets. The creation of tress, bushes and plants needs specific knowledge and demands a high amount of time, especially considering that forests often have a variety of species rising even more the time spent on the task. After some research online, the pack *Forest Environment - Dynamic Nature* from *NatureManufacture* was bought from Unity's asset store. The pack included thirteen models of tress, ten bushes, fourteen rocks and many more. It was also ready to be used in the High-Definition Render Pipeline, a necessity for the prototype.

Another huge advantage of buying these packs from the asset store is that they come game ready. More than being quality models already textured and evaluated in Unity environment, an important characteristic is that these assets come with Level of Detail (LOD) already implemented. LOD is a technique that reduces the number of GPU operations required to render distant objects (Unity Technologies, 2022). When an object is far away from the camera, it is noticed as having less detail than when it is close to the camera. However, Unity renders it with the same number of triangles at both distances by default, leading to wasteful operations, which can slow down performance.

An object must have a number of meshes with decreasing levels of detail in its geometry so the technique can be used. These meshes are referred to as *LOD levels*. The greater the distance between an Object and the Camera, the lower the LOD level rendered. This forces the 3D artist to create a different version of the same object for each LOD level, increasing the modelling time significantly.

The need to purchase another pack of assets was felt at a later stage of the prototype development process due to the approaching project deadline. The *Military Base Pack – HQ* from the creator *Halberstram Art* was acquired to be used in tandem with the custom-made assets of the area.

3.4.1.2.1 Hero Props

The assets grouped under the hero prop category were the ones that would need more time to be made as their workflow was the more complex of them all. It involved the integration of three different software: Blender, Substance Painter, and Unity.

The processes started with the creation of a low poly model. This model is the one that will be taken to Unity, so even if it could be framed as low poly, it cannot be so low to the point of losing too much visual detail. Taking the watch tower as an example, its model had 478 triangles in the greyboxing phase and 12.578 in this phase.

Aside from the modeling, a fundamental task to be done in this stage is to UV Unwrap the model before advancing in its creation. This process consists of the creation of a new set of information for the model’s surface. While information displayed in three-dimensional space — often represented by the X, Y, and Z axis — results in the shape of the model itself, a new set of coordinates — the U and V axes — can be utilized to create a plane in which the model's various surfaces can be flattened.

This flattening or unwrapping of the model’s surfaces into a 2D space is what allows the application of 2D images, or textures. This task is very tricky and time-consuming but should be done with attention, or the model can have continuity problems or even present distortions when textures are applied.

Figure 34 - A problem in the models UV made the texture on the upper front board to be rotated, breaking its texture placement.

An element that was not present in the initial phases but had to be made for the final prototype was the fuel filler gun for the gas station pumps. It was modeled over a reference image with the goal of capturing its silhouette and main shape for the low poly version. After that was done, a high poly version was made using the first as a starting point. This step allows that, later, on Substance Painter, it is possible to capture the high poly model's superficial characteristics and apply them to its low poly version in the form of 2D textures. This way, it is possible to imbue a low polygon model with a high polygon look.

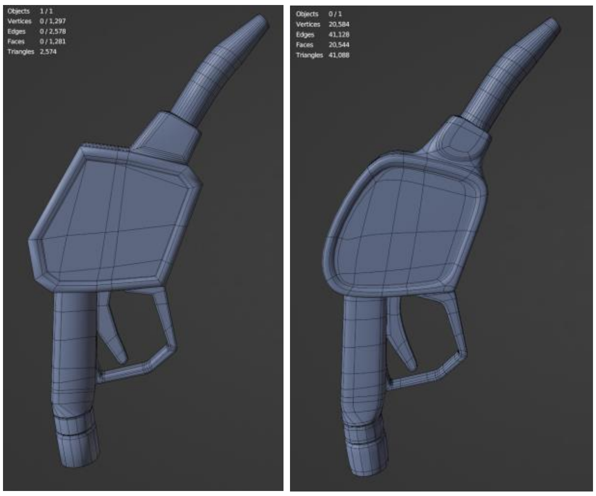

Figure 35 - Low poly model on the left with 2,5k triangles compared to the 41k triangles of the high poly on the right.

After having both versions modeled in Blender, a .fbx file was exported for each. The .fbx files are a standard file format for 3d models in the game industry, being the preferred format to be used in Unity and, also, in Substance Painter, the next software in queue for this process.

Inside Painter’s environment, the models were loaded, and the information from the high poly version was transferred to the low poly through a process called *baking* textures. After that, it was a matter of adding additional surface information such as color, roughness, metalness, and other details to make the model perceived in a photorealistic way. The software functions analogously to 2D painting software, having layers of information that can be stacked and masked to achieve the desired look.

Figure 36 - The gas pump model imported to Substance Painter.

After this process is done, Substance Painter allows exporting the designed textures in the desired file format. It is a matter of finding Unity’s HDRP option from a list of common output sources, checking if its settings are up to date with the latest changes and exporting the texture files.

Now it was a matter of connecting the model created in Blender with the 2D textures from Painter for the final assembly. To do that, the low poly file and the texture files were imported inside Unity. Prior to applying textures to the model, it is necessary to create a *material*. Materials are used in conjunction with other rendering components used in Unity and are an essential part in defining how an asset is displayed. Each material has a set of properties determined by the shader that the Material uses (Unity Technologies, 2022). HDRP has a standard shader that should be used for photorealism. It has a list of material parameters from which it is possible to recreate the look of most surfaces.

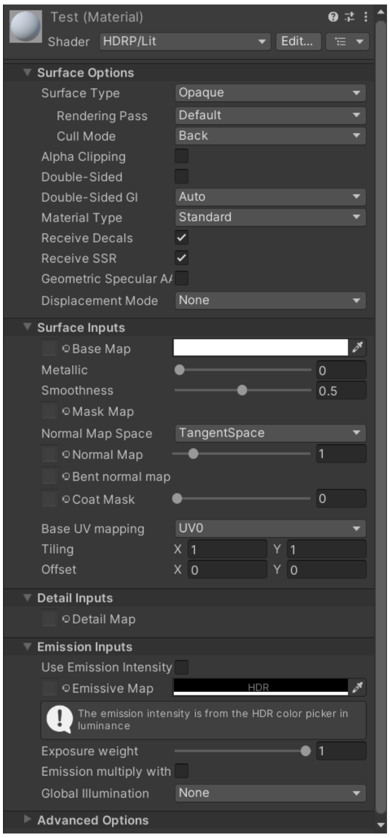

Figure 37 - The inspector of a standard material showing its adjustable parameters.

The textures are placed on the material, which is then placed on the model. This ensemble can be saved as a *prefab*, an element that stores all information for future use.

Figure 38 - The fuel pistol complete as a prefab in Unity.

The hero prop pipeline was used in smaller assets and the ones that would appear closer to the camera, as was the case for the collectables, or when the asset was something with a complex superficial characteristic, as was the case with many of the gas station elements. The gas pump, fuel filler, and store all had varied materials and graphics on their surface. The gas roof structure that covers the pumps was just too unique to find an *out-of-the-box* solution and also had to be made with this process.

This is a *high risk, high reward* kind of pipeline to be adopted because, even with the reliable results it delivers - being able to deliver as photorealistic results as the skills of the one who makes them allows - is a process with many bottlenecks. It demands a lot of attention in the UV unwrapping as Substance Painter baking texture needs perfect UVs to work correctly. The texturing process alone is something very time-consuming as one can spend as much time possible to achieve the intended results.

3.4.1.2.2 Environmental Assets

This is the group that includes all bigger assets, buildings and objects that had only a simple texture applied to them. This category followed a different logic, especially regarding its use of textures, than the hero prop one. Here, the pipeline used was what is called *texture tiling*, meaning that the applied textures would repeat as many times as needed to cover the model's surface. This method offers advantages as it is not limited by the resolution of the chosen texture, so if the game camera ends up close to the surface, it would not appear blurry or pixelated. It also enables the possibility of using small resolution images, delivering sharper results with less computational power use. The negative side is that it is easy to end with an uncanny result, non-photorealistic or, at least inexpressive or boring.

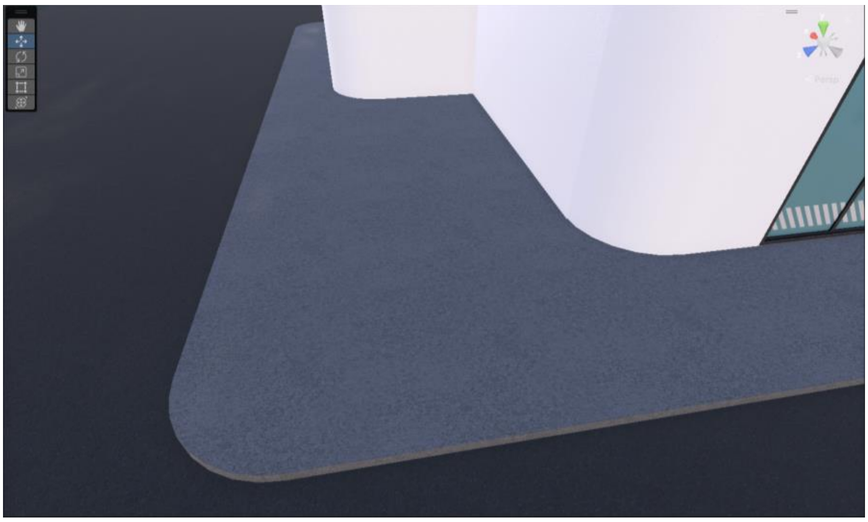

Figure 39 - Flooring of the gas station store. It is possible to perceive the repetition of the texture in the pattern.

Two things can be done to mitigate this matter. The first is a well-thought UV unwrapping. All curbs and other ground elements on the scene utilize the same concrete material. Apart from their simple geometry, these models had their UVs planned in a manner that would allow them to be used with a generic material. Concerns related with size and proportion of the UVs and rotation so the texture would flow consistently through the geometry.

The second way to lessen the texture tiling disadvantage that was experienced was blending two or more textures in the same model. This is a process commonly used by environment artists as it can overcome the problem completely, if properly applied. As this was an unknown technique, its application was assessed on an asset before being replicated. The cabin if the starting area was chosen as it is the first building the player can interact with, and its look should be captivating.

The technique was explored inside of Blender, as this software is more familiar to this author. After the modeling was finished and the UVs properly unwrapped, it was a matter of adding the textures to the model. The blending is controlled by a mask, consisting of a hand-painted image also applied to the model. Each of the image's color channels controls a texture, so by painting, for instance, the color red on a wall, the texture A is applied, while painting green would display the texture B in its turn.

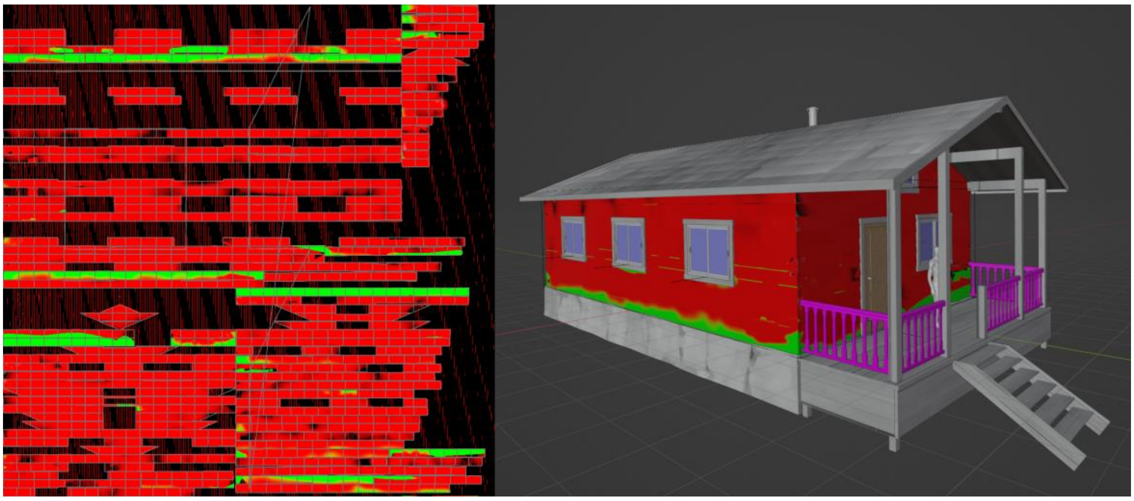

Figure 40 - On the left: the flat blending mask texture. On the right: the mask applied to the cabin walls.

To wield satisfactory results, many parameters had to be tweaked using Blender’s shading properties, which was impossible to take to Unity as it was later found. The solution would be to bake the changes made in the mask, so then it could be sent to Unity, but a way of doing it was not found. A solution would be to re-create using Shadergraph, the parameters used in Blender, but, as straightforward as it sounds, this idea came only after the project was concluded. The prototype ended up using the raw painted mask resulting in sub-optimal results. Due to this matter with the blending mask, this process was not replicated in any other asset. The texture tiling pipeline turned out to be employed in its most basic form.

Figure 41 - The end result for the cabin asset inside Unity.

A case that ended up using a method of its own was the building structure presented in the military base area. This asset was constructed as a modular set to be easier to transport from Blender to Unity and to allow some iteration between the modules. As the modules would share the same materials, it seemed correct to adopt the use of a *texture atlas* for those models, this way, they all shared the same material inside of Unity. A texture atlas packs more than one material in the same image so different models can share it. To safely use it is a matter of adjusting the UVs of each model to its proper position in the image.

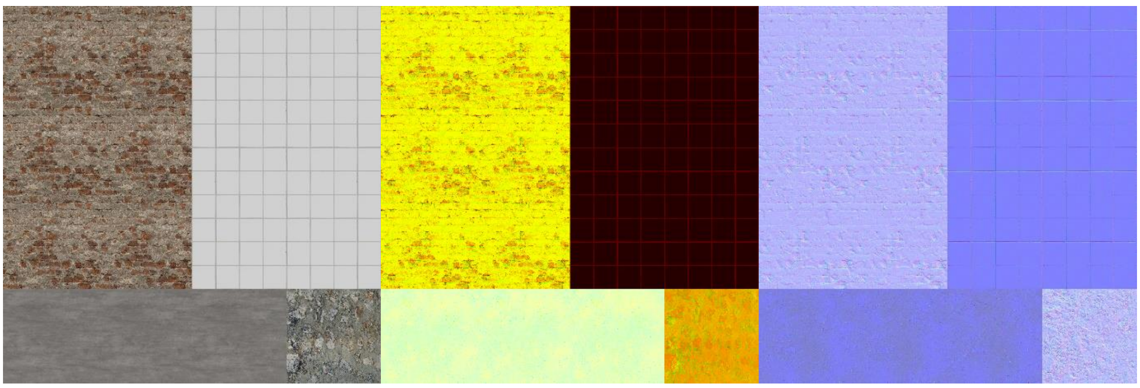

Figure 42 - Base color, mask map, and normal map of the texture atlas used for every module on the building structure.

The downside of this technique is the opposite of the texture tiling in the matter of image resolution. To ensure that the image is not pixelated, the textures end up being large, as models use only a part of each texture. As the performance expends of large images is high, it is important to know when to use texture atlases. In this case, the asset was composed of more than two hundred modules, so using the technique felt justified. Thus, as a result of the applied technique, the brick walls ended up looking repetitive as there was many of the two hundred modules were brick walls. More research has to be done in texture pattern on repetitive models to reach a suitable result.

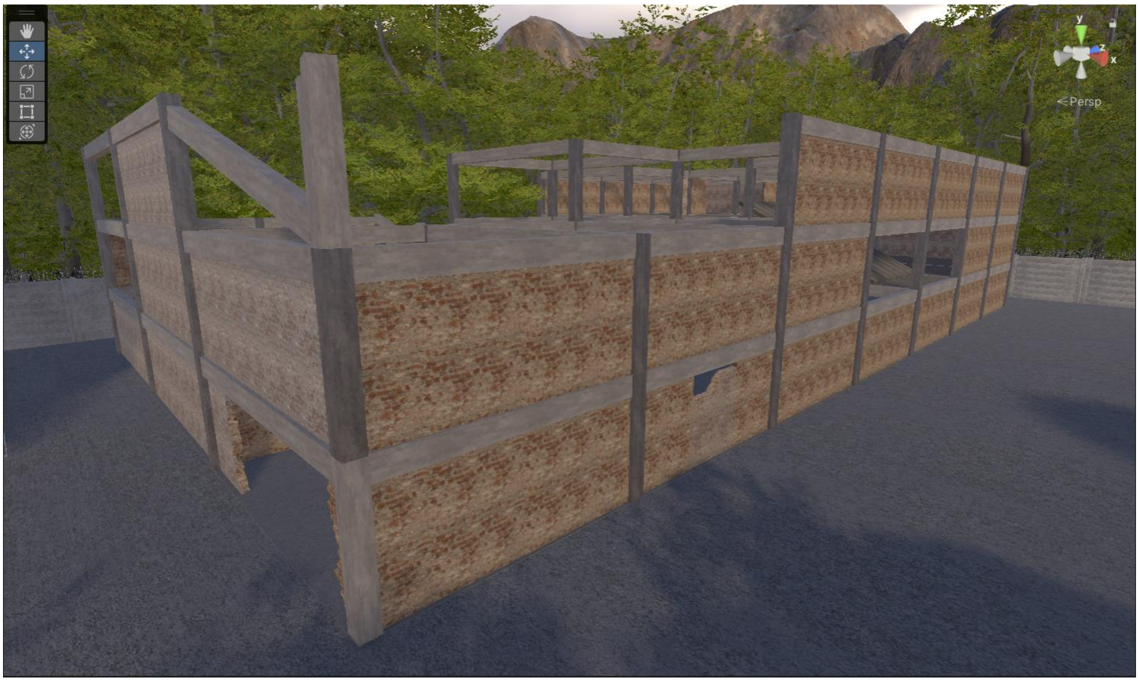

Figure 43 - The repetition of the brick texture is noticeable in the final version of the modular building.

In total, 134 .fbx files were created, not counting the purchased assets. This process ended up not being a linear one. It involved some iteration and back and forth between the three software environments to adjust the assets. It was a stage that required much time and work to be finished, especially the texturing part. Comparing the time spent modelling with the time creating textures and materials, the latter occupied double the time than the first. This was something unexpected at the beginning of the project, as texturing was not accounted for as a bottleneck in the project. Asset creation started as a process done on its own but ended up being finished alongside the assembly of the environment.

3.4.1.3 Assembling the Environment

In this stage, the provisional models of the block out stage are replaced by the final models to shape the 3d space of the game in its final form. This task is the responsibility of environmental artists in a digital game studio as it represents the bulk of their work.

The process is followed by the prototype's environmental areas, advancing, initially one by one and adjusting everything together in later stages. The forest area development began by placing the trees according to the definitions from maps and concepts. The Terrain tool has a function to spread prefabs in the desired location of the terrain, but it presented a conflict with the models from the already bought pack.

Another tool had to be found to make this placement, as adding it one tree at a time would exceed the time constraints of the project. After experimenting with some available options, Prefab Brush + ended up being bought as it proved to wield the required results for this task. It enables the placement of different prefabs at the same time via brush-like mechanisms that deposit them on the desired surfaces.

Figure 44 - A circle appear in the cursor position to indicate the area where the chosen prefab will be placed.

For the forest to have a more photorealistic look, an abundance of trees, bushes, and plants, both in terms of volume of elements and in their variety, is a necessity. This abundance in volume and variety of assets have a high-performance cost making the prototype heavy and slow to work with. Optimization would be important to make this prototype into a final product.

This same need for abundance requires the environmental artist to spend a lot of time detailing the scene, as much of the photorealism comes from this detailing. Taking as an example a natural forest, an observer can see trees of different shapes, heights, and shades in it, even if it is the same species. A photorealistic scene is expected to emulate the same kind of experience for the player. Furthermore, the arrangement of elements in space is not predetermined and fixed, a modular repetition can be perceived and cause detachment or uncanniness, so this task is better done manually. This means a huge time consumption as the environment is not small and demands a lot of attention.

The starting area, which includes the cabin and its surroundings, was one that demanded special attention as it would be the first game area the player would see. The cabin was a matter of simple replacement between the interim and final model. The setting was later completed with the addition of vegetation. Only minor tweaks to the landscape shape and texture were required to adapt the elements, but no significant changes were made.

Figure 45 - The cabin area finalized.

For the gas station was applied the same method as the one used in the cabin area. First, the placeholder models were substituted by the final ones. To emulate the symbols and line signs on the floor often found on this type of establishment, Decals had to be employed. Decals are materials that decorate the surface of objects that already have material applied to them (Unity Technologies, 2022). They are most commonly used for effects like paint splatters and stickers and would fit well in this application. Decals work as a material projector, displaying their graphic information on the surfaces that intersect with them.

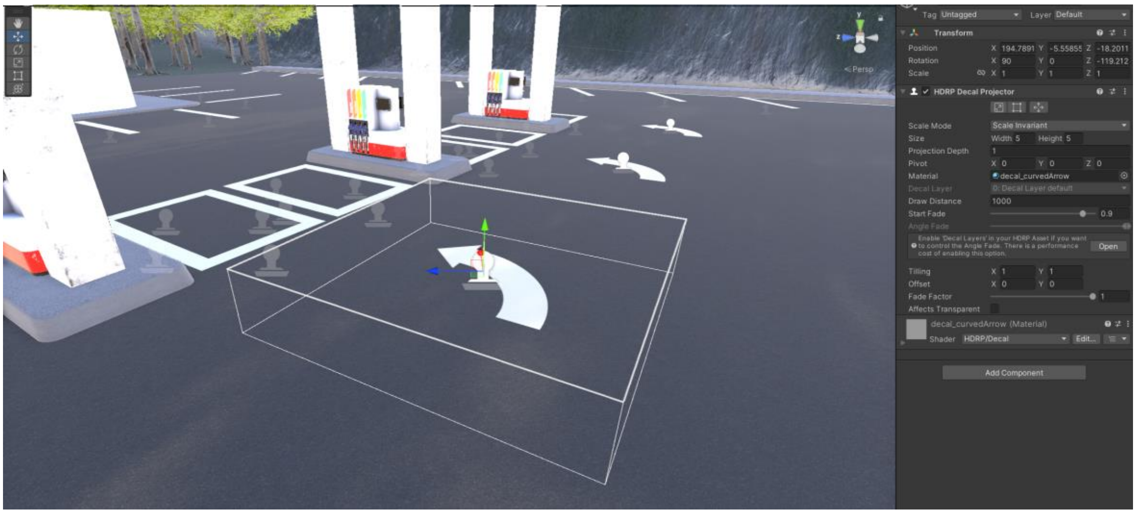

Figure 46 - The decals are market with a unique gizmo in the shape of a stamp.

The road did not allow substantial changes in geometry, so the only thing that could be done there was the addition of textures to simulate the asphalt look. This impossibility is due to the geometry creation tool used in the first place. A deeper understanding of programming Bézier curves and splines and making them control geometry was necessary to insert more complex geometry, such as a guardrail, curbs on the sides, the road, or speed breakers at time intervals. Another solution would be to buy a ready-made tool like Curvy Splines 8 – available in the asset store - but even though the addition of these details would increase the photorealism of the scene, the gain in scope would not justify the budget investment, so the prototype would remain with its initial solution.

In the greyboxing phase, only the largest assets were placed in the military base area, as this was enough to visualize what this space would look like. With the exchange for the final models, the fact that this environment lacks more details became explicit. This was already the area that demanded the most in terms of asset creation but as the space was large, the need to populate it with more elements was clear, so the purchase of a pack of assets was justified in this case.

Working with asset packs is not free of surprises as the quality of the assets is not always clear before the purchase. In this case, many of the assets need retexturing due to compatibility issues with the render pipeline.

One element that would be effective would be the inclusion of debris throughout the area. For the creation of this category of element, there are specific software that deal with destruction simulation and the physics of materials, such as Houdini. Blender has a section dedicated to physical simulations. However, these simulations demand specific knowledge that, to be acquired, would require time. As there was not much time left at this stage, no debris was included in the scene.

3.4.1.4 Lighting Settings

One of the advantages of working whit the High-Definition Render Pipeline is its capacity to produce physically based lighting and materials (Unity Technologies, 2022). It uses high computing power to produce results, so it is intended for higher-end platforms. Its main advantages come in the shape of its convenient post-processing and volumetrics capacities, shadowing quality, and the already mentioned physically based lighting. All these qualities were explored in this prototype to ensure that the photorealistic look was achieved.

Another occupation from the digital game community can be pointed out regarding lighting matters. The Lighting Artist works exclusively with light and rendering. This occupation is highly specialized and usually is available only on AAA game projects.

The HDRP uses a *volume framework* (Unity Technologies, 2022). To begin, a *volume component* was added as a dedicated GameObject to Unity's scene so lighting settings could be managed. A volume profile will store the changes done via a mechanism called *override.* Each override is a new setting that alters default illumination properties, such as Exposure, Bloom, Sky and Fog.

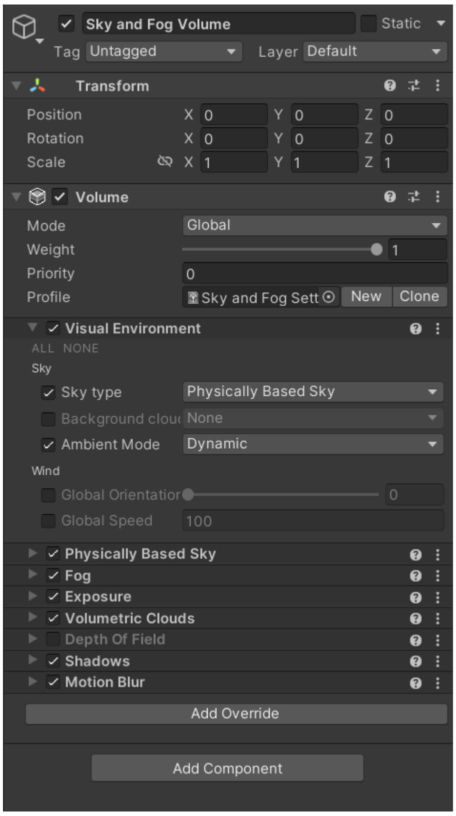

Figure 47 - The inspector shows the Sky and Volume object with several overrides.

Taking advantage of the fact that the environment is external, the photorealistic scene used volume overrides to create a sky simulation that could provide all illumination needed for the entire scene. Besides using the Physically Based Sky override, it also uses the novel volumetric clouds. This override was recently added to Unity and is responsible for the creation of clouds that can render shadows and receive fog and volumetric light. Another volume override that contributed to the illumination of the scene is the Fog Control, which is responsible for dispersing fog on the environment, making light more diffuse and obfuscating far from the camera elements. The combination of this feature with the use of a single source of light to function as a sun is responsible for all the illumination effects of the scene.

Apart from the aforementioned overrides, Exposure was the only other setting adjusted to contribute to the final look of the scene. Depth of Field was experienced with, but the results have not been adequate, so the effect was ruled out of the prototype.

Another matter that contributes to the high-end look of today's photorealist games is the use of Anti-aliasing methods. Anti-aliasing is the smoothing of the jagged appearance of diagonal lines in a computer-generated image (Peddie, 2013). It was only used in non-real-time renderings as the computational cost can be high, but more recently, games have already made use of this feature.

HDRP provides many approaches to anti-aliasing, such as MSAA (Multisample Anti-aliasing), which provides the best results but with elevated costs. Temporal Anti-aliasing (TAA) was the chosen option for this prototype as it provides a compromise between performance and visual quality.

Figure 48 - With a magnification of 3.5x, it is possible to compare the jaggedness of the object's edges from an image with TAA (left) and one without it (right).

It is easy to understand how illumination can be an occupation on its own. The process of adjusting lighting settings is a finicky one. The results achieved after many days of adjustment were acceptable but could not meet expectations. The outcome was below the potential of the chosen render pipeline due to a lack of experience with its tools.

3.4.2 Stylized Level

3.4.2.1 Defining the Visual Outlook

As the stylized scene would be made with the photorealistic as a base, its development had to be delayed until the final portion of the prototype phase. A test scene was constructed as an experimenting area for the stylized look in order to anticipate some of the challenges that could arise during production. This enabled the design of a stylized shader, which is the essential defining feature of the complete styled appearance.

As mentioned earlier, shaders are algorithms used to describe how materials and light behave in a computer-generated image. The shader algorithm is what determines how an image will be rendered and displayed to its viewers. In a digital game studio, shader creation stands in between the programing and art department, as programing knowledge is necessary to build them while its parameters will impact the game's appearance. The role of the technical artist was created as both artistic sensibility and technical expertise are necessary to fit the task of bridging this gap. Their main job is to streamline how art is made and integrated into the game (Riot Games, 2018b) and shader development is one of their tasks.

For the prototype stylized scene, the goal was the creation of a master shader from which all material could be derived from. This strategy was beneficial since it encouraged the development of a single shader on which all of the efforts could be focused. It also helped with visual cohesion because all materials would follow the same mathematical logic to be rendered.

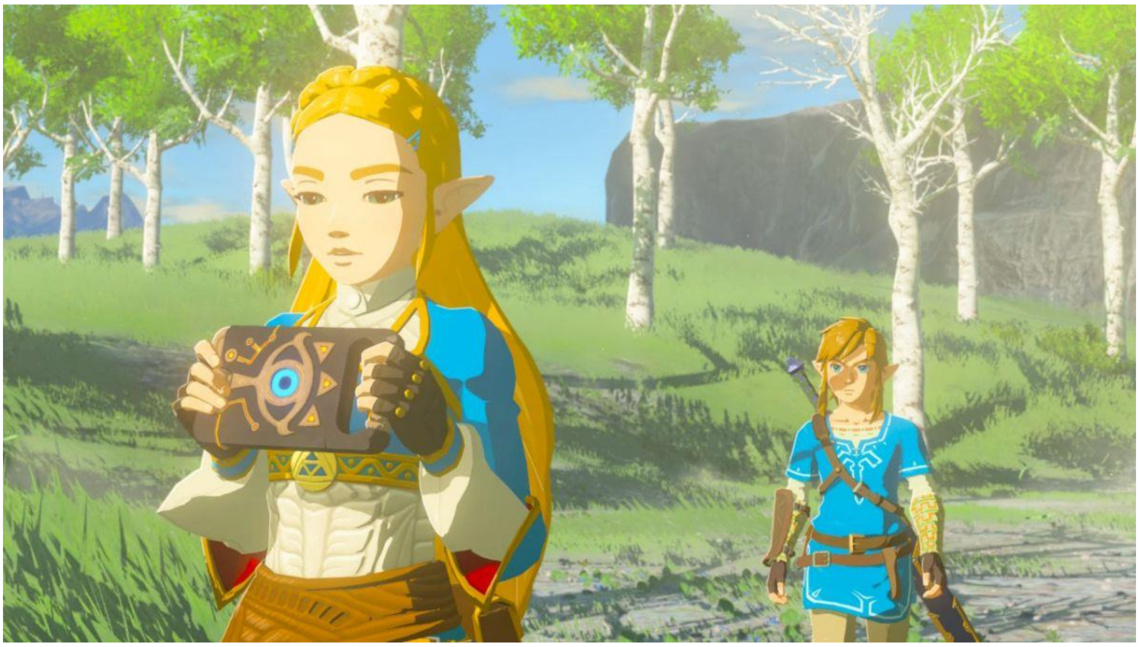

Figure 49 - *The Legend of Zelda: Breath of the Wild,* 2017 (Loveridge, 2022) is a game that uses the cell shading technique.

Many different appearances can be framed inside the stylized look, as discussed in the previous chapters. The chosen appearance for this prototype was the cartoonish look of the cell shading technique. This look is a staple used in many games and would fit the familiar/generic premise stipulated at the beginning of the project. The look would be achieved by composing elements, namely by flat shading, whit 2 or 3 tones variation instead of a gradient one; a stylized version of light reflection instead of a physically based one; and line contours in the edge of the objects to complete a cartoonish appearance.

3.4.2.2 Creating the shader using Unity's Shadergraph

Shader Graph is Unity's visual editing tool that allows the creation of custom shaders with no or little coding experience and is the tool of choice for creating the stylized master shader. It works via interconnecting nodes in which each node is responsible for a function inside the algorithm. These functions can be simple math operations or assume a more complex role, such as accessing the object UV information or the game camera position.

There is an immense volume of content regarding how to make shaders in general and more specific ones for how to use Shader Graph within Unity's community and many of them were analyzed prior to starting development. Ben Cloward's YouTube channel ended up being fundamental for this project as it provided information grounded in experience, since Ben Cloward has been a game developer for over twenty years and now works for Unity on the Shader Graph team.

An obstacle that happened right from the start of the shader creation process had to do with the need to access the game Main Light information inside of Shader Graph. This was important because without it is impossible to control how light will behave and the shader ends up having to use a standard process.

Many sources indicate that this was not possible to be achieved using shader graph and could only be done by writing HLSL, Unity's shader programing language. In one of Cloward's videos, a solution is presented in Unity's 2022.1.0b14 beta version, in which the node containing the function to access this information is already there, allowing Shader Graph to still be used. The downside of it was that for it to work, the whole game project structure had to be ported to a beta version of Unity. Beta versions are more prone to failure, have bugs and don´t have any support in case the project breaks. Taking these risks into account, the option to proceed with the transition to this version was cleared since the project files could be backed up in the previous version in the state they were at the time and, in the event of a problem, these files could be restored.

Following the creation of the shader, its structure can be broken into various parts. The first is responsible for calculating the overall illumination of objects. This was accomplished by writing a code that stated that the point of the object facing towards the light would receive full illumination, while the point facing away would receive no illumination. The in between points would receive a fraction of the light, depending on their position, forming a soft gradient between the point of maximum illumination and the point of no illumination. This model is a Shader Graph adaptation of Lambert's general illumination law described as follows:

$$I_d = \frac{k_d I_p}{d^2} \cos(\theta),$$

where k_d is the diffuse reflection coefficient, I_p is the light source intensity, d^2 is the square of the distance of the light source to the object's surface, and θ is the angle of incidence of the light source.

To break the generated gradient in solid tones, a 64x64 pixel image serves as a guide to the tone mapping on the object when combined with the first part. This image holds many strips, each with one type of stepped tone shading. By cycling vertically through the image, it is possible to select which kind of shading is displayed. This solution gives versatility and adds some variation to the master shader as the kind of shading could easily change for each object inside its material settings.

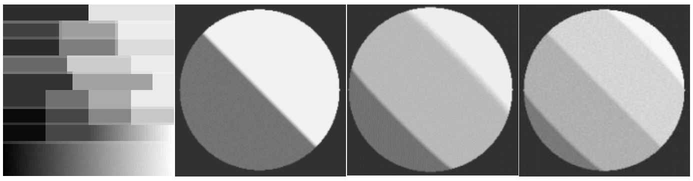

Figure 50 - the horizontal strips have different tone maps allowing various results as displayed in the preview spheres above.

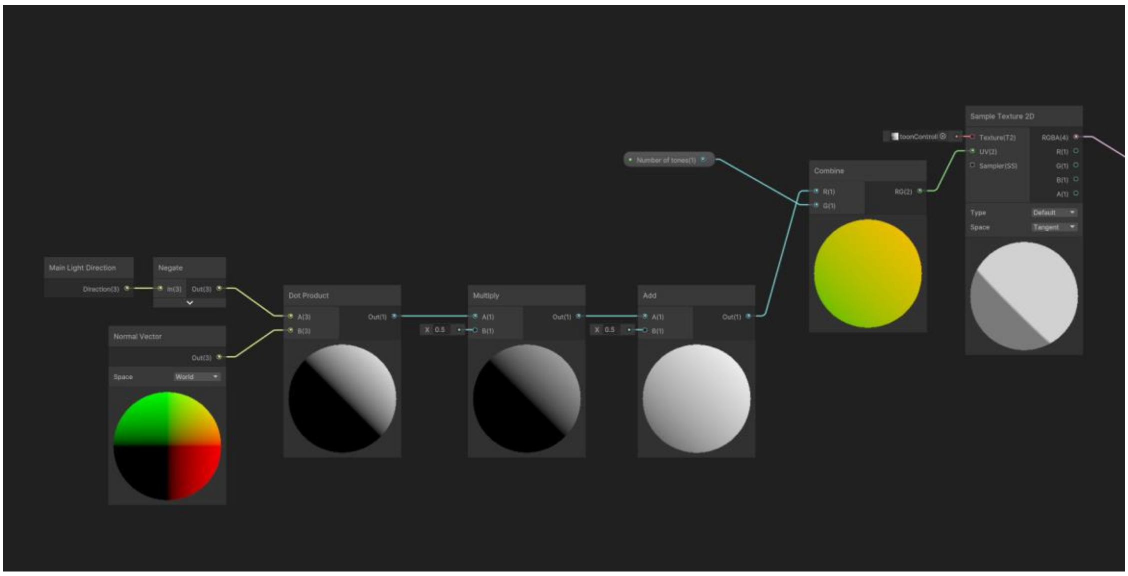

Figure 51 - the section of the shader depicting the basic illumination process and the addition of the stepped gradient modifier.

The code that comes from the above section will provide the Base Color for the objects that use this shader, but it still needs more elements before it does. Color is added by multiplying its information with the previous outcome so the, until now, greyscale could be tinted.

While the shader was being prepared in a test scene, this setup provided all Base Color information. However, after the migration of all elements from the photorealistic scene to the stylized, the lack of texture information was missing for some models, especially those that made use of the alpha channel for transparency. To solve this issue was a matter of multiplying the outcome again, but now with 2D texture information.

The initial idea for the shader was to use only pure colors and outlines, so using textures was necessary but undesirable. However, the addition was beneficial not only to transparent assets but also needed for objects with abundant graphical information, such as the majority of the assets that used the hero prop framework.

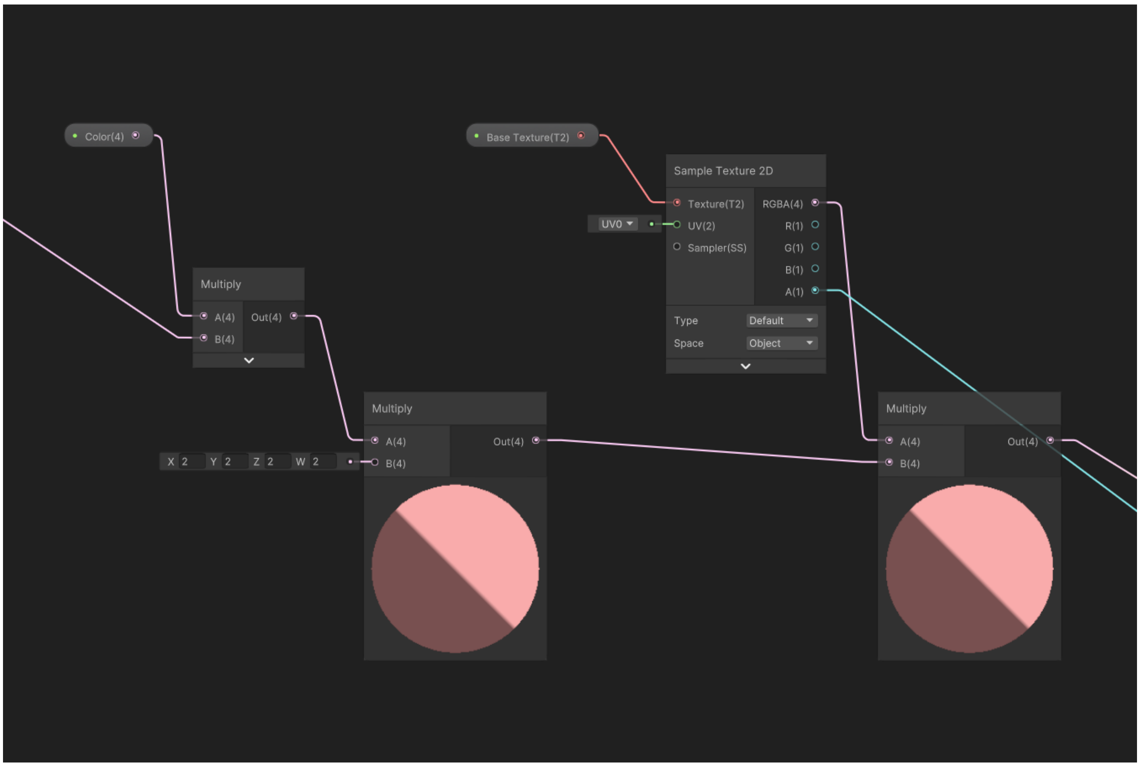

Figure 52 - the section of the shader depicting color and texture information being added to the mix.

The next element added to the shader was some pen strokes highlight to amplify the cartoonish look. These were achieved by a combination of similar elements used in the previous parts of the shader. The object point facing light was calculated and multiplied by a pen stroke image positioned in that same position as to fake specular reflections. Many parameters of the pen stroke can be controlled to continue with the *one shader fits all* approach.

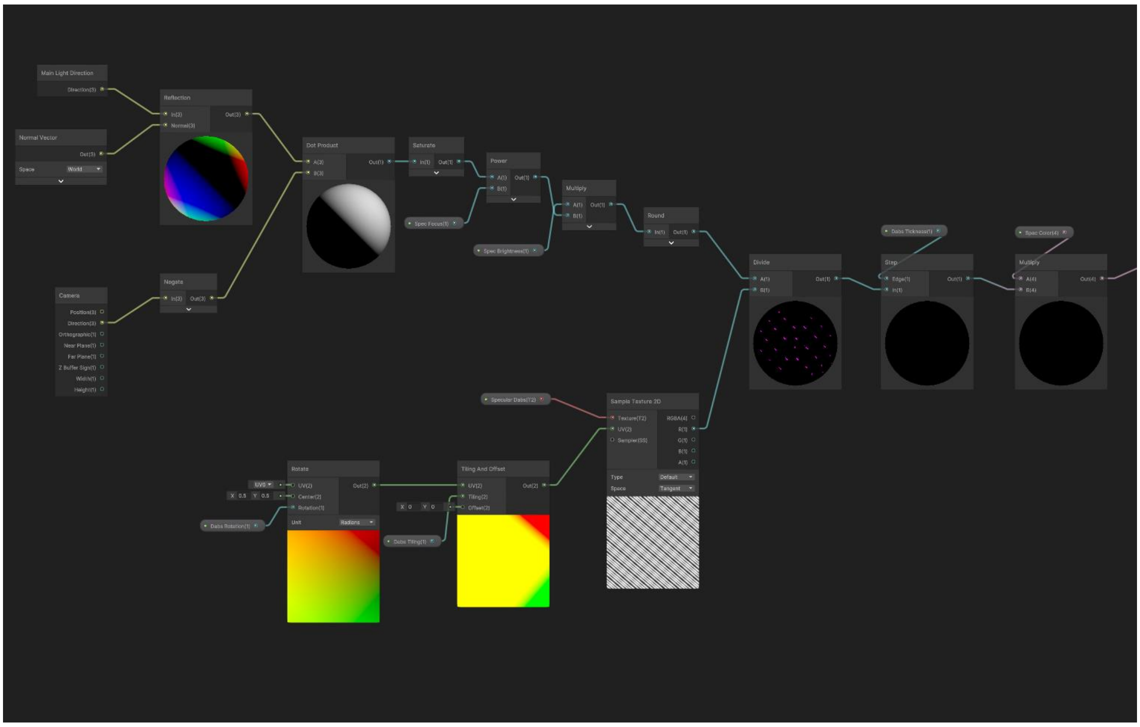

Figure 53 - pen strokes highlight section of the shader.

The only missing element to complete the shader's initial design was the outline contour. This proved to be a more significant challenge than first estimated and ended up not being implemented. One of the most traditional techniques to achieve this feature, used in games for decades, is the ingenious use of a duplicated mesh for each object. This mesh is scaled to be slightly larger than the original as it is responsible for rendering the object's outline. As its use was not planned from the beginning, changing all the models again would be an overly complicated task, making the use of the technique unfeasible.

Other techniques were discovered, but all started from the same principle of doubling the object mesh. A promising method that made the creation of surfaces possible was to use geometry shaders. These shaders can not only control the behavior of rendering shapes but also control their creation. This way, the process of duplicating the object mesh could be done inside the shader instead of necessitating going back to the 3D modelling software. *MinionsArt* GitHub page and YouTube channel are a major source of information on how to use geometry shaders and could provide a straightforward way to implement the solution that, unfortunately, had no effect within the project. Due to a lack of documentation on the beta version the project was being built in or an unacceptance from the HDRP of using this kind of shaders, this route had to be abandoned too.

There are some outline tools and shaders available in the Unity Asset Store. One of them, called *Highlight Plus* by *Kronnect,* had compatibility with HDRP and was bought as a last attempt at the desired feature. The tool worked fine but only when using HDRP basic shader, so its use in tandem with the stylized shader was not compatible, and the option had to be abandoned as well.

As time passed, the outline feature had to be discarded to comply with the project schedule. Even without one of the initially defined features, the shader results were satisfactory in its design. It was able to be applied in all elements of the stylized scene with the exception of the terrain that had its own settings.

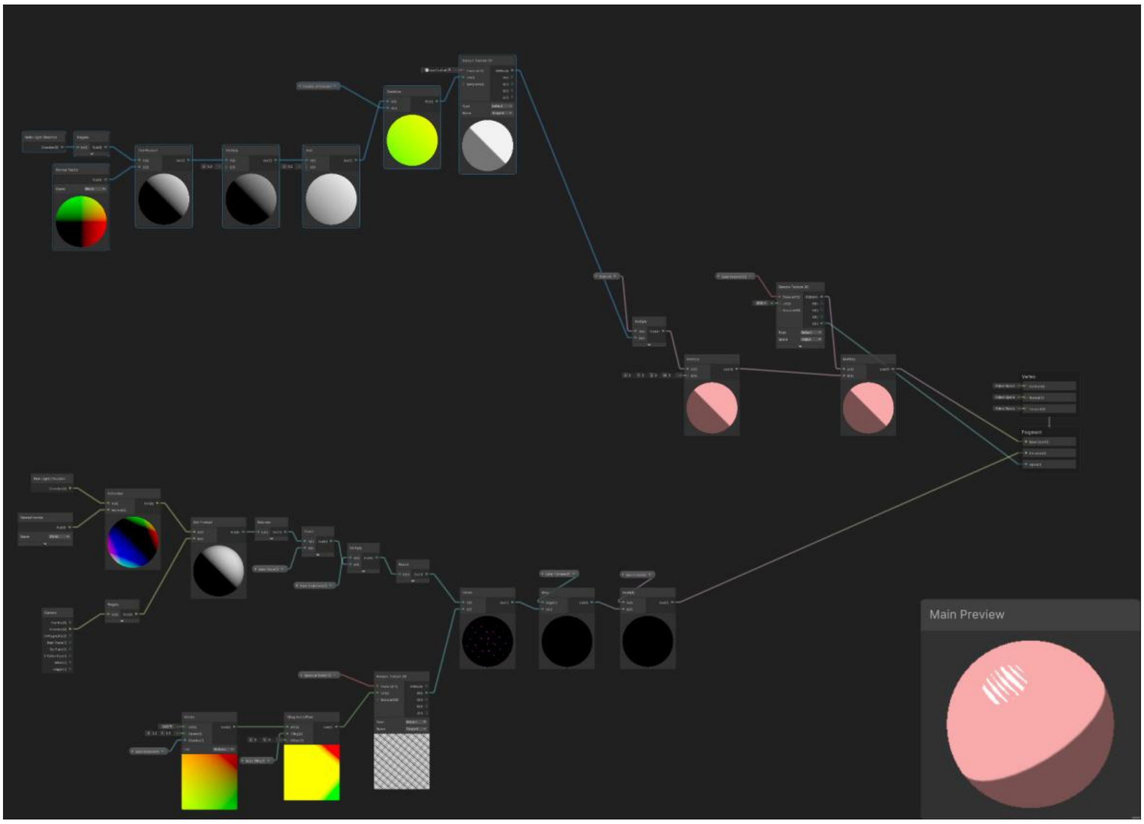

Figure 54 - the final shader graph code for the stylized scene.

3.4.2.3 Turning the Level Stylized

Until this point, a test scene was used to fiddle with the stylized elements created. The styled level could only be made after the entire photorealistic scene had been completed, including all environmental features, by copying the scene file such that the environmental information was identical.

Following the steps of the photorealistic level, the terrain was the first element to be altered. In this case, the topographic shape would remain the same, and only the textures would change. Unity's terrain holds its information in a *Terrain Data* file that had to be duplicated so the textures in this scene could be changed without altering the previous. In this transition, the texture information of the photorealistic scene was lost and had to be redone a second time, as the existence of this file was unknown until now.

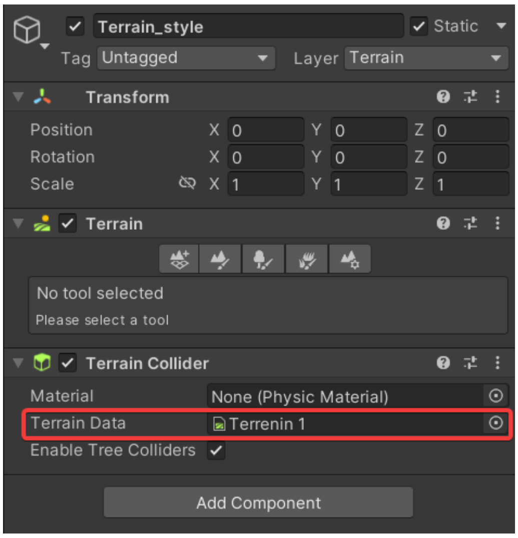

Figure 55 - The Terrain Data will hold information not only regarding collision but also texture maps.

After dealing with this issue, a set of simple 2D textures was created in *Affinity Design*, an illustration software. In this case, only base color information was necessary, so there was no need to use software like Substance Painter as the other texture maps would not beneficiate the visual outlook.

Figure 56 – stylized terrain textures – stone, dirt, grass and path.

The created textures are abstractions of the photorealistic materials from the previous scene. Four materials were chosen to compose the stylized terrain as they were more significant in defining the terrain typology as seen on the photorealistic level.

In the sequence, all assets imported from the photorealistic scene should have their materials changed to their stylized version. This would be a strenuous process if, for each asset, the change had to be done manually, so a tool was created to swap game objects of a kind from the scene for the desired one. This way, a single asset could be prepared, and then all other instances could be switched by this one.

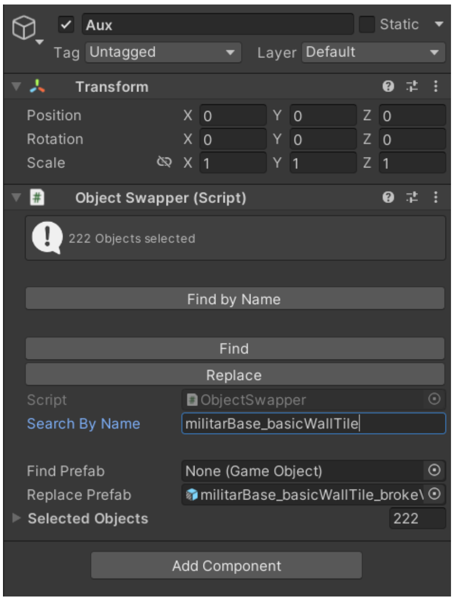

Figure 57 - Inspector of the Object Swap Tool.

Thus, the process of adapting the 3D assets to the stylized look was streamlined as follows. First, a copy of an asset was added to the scene, and then a material was created for that asset. The stylized shader provided the necessary settings so each material could be configured for the asset need. In the majority of cases, it was a simple matter of adjusting color and shading tones but, due to the geometry of some objects, more settings could be adjusted.

The concern was that this asset with freshly applied material would conflict with the previous one in the photorealistic scene. However, assets, unlike terrains, only exist in the context of the scene and would not clash. Secured by this information, now it was time to put the asset exchange tool to use. The tool worked seamlessly, making what could be a time-consuming task something effortless.

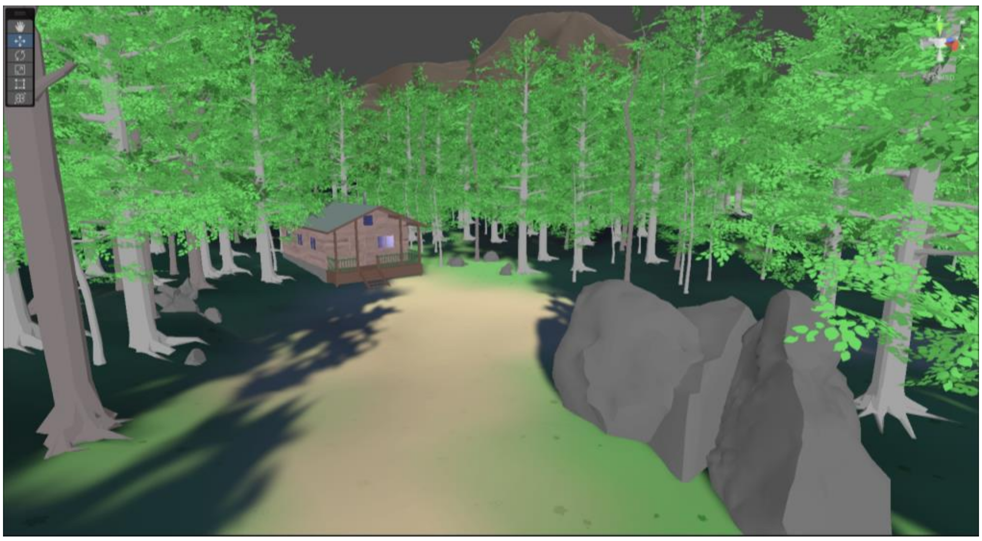

Figure 58 - All elements for the cabin area were changed using the swapping tool.

An element that was not swapped in the above process was the small vegetation and underbrush from the photorealistic level. To replace these, the intention was to use a single grass asset following the concept of simplification of elements. The first attempted method of filling the scenery with grass was by using a geometry shader, but as described in the stylized shader section, HDRP cannot process this kind of shader, so another method had to be devised. The chosen framework ended up being the same used for the other props, i.e., a 3D model with a custom material dispersed with the Prefab Brush + tool through the terrain.

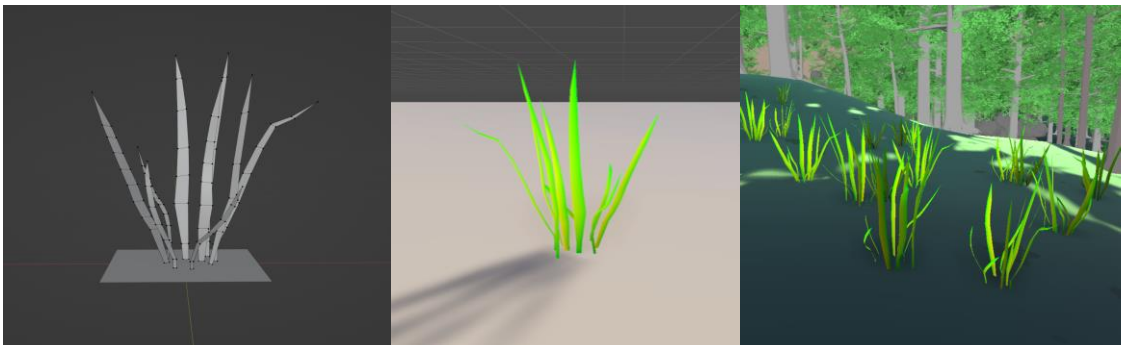

Figure 59 - Grass module model on the left, the prefab in the center, and a grass patch in the scene.

The intention with this grass element was to fill completely the terrain area using the grass texture, covering most of the ground. But even creating a simple model – composed of only 120 triangles – and using an easy-to-compute shader, dispersing a huge amount of grass in the environment proved to be impossible to achieve. The high amount of grass elements even broke Unity's game project completely, forcing the use of a previously committed version from its git repository.

In the end, the grass asset could be used but only sparsely through the terrain. This is a case that illustrates the difficulty of creating a digital game by a single person. This, clearly, is a graphics optimization problem that could be solved by a programmer with experience in creating three-dimensional games and dealing with this type of situation often. In the context of real-time rendering products such as a digital game, art and programming have to walk hand in hand to achieve the desired results in a suitable manner.

3.4.2.4 Rendering and post-processing effects

The last elements to be explored were the ones connected to the final rendering of the level. Different from before, the realistic lighting of HDRP was not necessary as this level would not benefit from it. The illumination settings were lowered to a minimum, removing the volumetric clouds, the sky effect and maintaining only the sun as the main light focus – as this was necessary for the stylized shader to work.

Inside the camera settings, even the standard sky was turned off to be replaced by a solid blue color. Antialiasing was also disabled entirely, resulting in a choppy silhouette that contrasted with the photorealistic solution.

Post-processing effects were identified as an interesting tool for achieving a unique and interesting visual to contribute to the development of the stylized look. These effects work via the same volume framework mentioned in the photorealistic level illumination section. They are a kind of override added to the scene in the same way as any other volume override.

Apart from image adjust like Exposure and Depth of Field, the stylized level could benefit from a custom post-processing effect. These effects can be created in the same way that shaders are, but they require far more programming knowledge than is required to use Shader Graph. A survey had to be conducted in search of sources of information on how to proceed.

There is much online content about shaders but only a few about HDRP post-processing effects creation. It is believed that this result comes from the novelty of this tool. There was not enough time for the Unity users to explore it, generate and make content available for the community. The information that was found was aimed at programmers, so there was a barrier to entry to actually accessing it.

A single source was identified as having content that could be accessed and also applied to the prototype. Zgeb (2021) presented a comprehensive tutorial on how to create a pixelating effect with post-processing effects. The tutorial was followed without significant issues, and the effect was implemented on the stylized volume override stack, integrating seamlessly without any significant conflict. The effect's intensity level can be controlled to fit the desired look but, after its implementation phase, it was decided that the effect did not benefit the visual outlook.

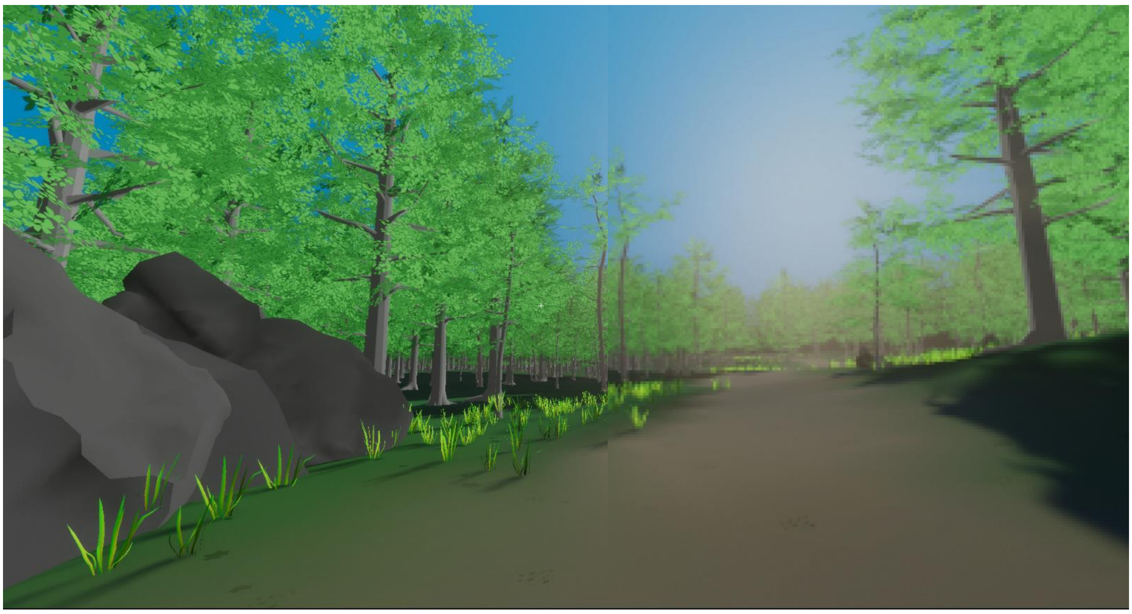

Figure 60 - The right image has the pixelating effect while the left remains without it.

The stylized scene lighting setup ended without the addition of any custom post-processing effect, but this is a topic to be more explored in the future as its capabilities seem promising in adding non-photorealistic effects and contributing to the overall look of a digital game. The stylized scene lighting setup, as well as the scene itself, were completed. With both levels complete, it was time to move on to implementing the game mechanics to tie both levels together and finalize the prototype.

3.4.3 Game Mechanics

It is likely that the game designer will be in charge of writing the necessary code to implement the intended mechanics in the framework of a prototype because the quality of the code in this stage simply needs to be good enough to work. The game designers are responsible for creating the mechanics, but the programmer or engineering team is responsible for implementing them in the definitive version of the game, as all the code needs to be refactored to contain the necessary information to ensure its proper functioning, performance, and communication with external systems.

To build the scripts necessary to create the prototype's mechanics, much was drawn from the developer community around Unity, as the necessary knowledge to write C# scripts was not an established skill. The designed mechanics are simple and should be quick to build; however, due to the low skill level, they took a long time to execute correctly in the game engine.

3.4.3.1 Player movement

The mechanics proposed for this prototype, as mentioned in the Understand and Define phase, were simple enough to allow the space produced to be experienced in a game setting. The first one to be implemented was the first-person navigation perspective. Initially, we used a standard system provided by the game engine, so early, the option to walk on the developing level was already possible. This was important because assuming the same camera position as the player helps debug flaws that were not observed in any other way.

Later in the development, a custom navigation mechanic was implemented to have more control over its settings. A *jump* and *run* functions were added as those are expected in games with similar navigation as the prototype. Settings such as walk speed, run speed, jump heigh, and gravity can be tweaked to the desired results.

List of player movement controls

- Move – W/S/A/D keys
- Look around - Mouse
- Jump – Space bar
- Run – Left shift + W/S/A/D keys

3.4.3.2 Monologue system

The second game mechanic to be developed would be the monologue system. This system would be in charge of displaying text in pre-defined places and moments of the game. This was intended to serve as a guide in the exploration and serve as a motivation to continue to play.

This system was developed using a series of triggers that would be activated when the player entered its area. Each trigger had its own text associated, organized by a chosen number of phrases that corresponded to the part of the monologue to be displayed. The phrases would prompt in a text box created using Unity's User Interface elements. A Manager script controlled the connection between these elements and all the logic of the system.

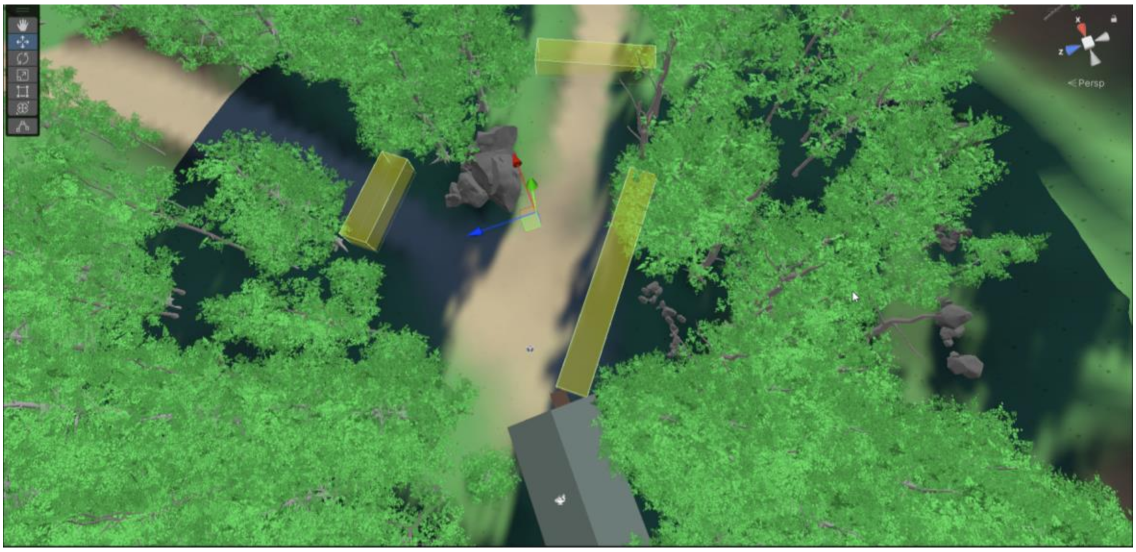

Figure 61 - The triggers were identified by a colored box in the scene editor, so their area of effect was easily controlled.

Each level had its own group of triggers scattered in the environment. They had to serve the purpose of conveying some information to the player, but the order in which they would be accessed cannot be controlled as the space is open, and each area can be accessed from different directions. Thus, the text they displayed had to be open enough to be understood within a larger, non-linear context.

This non-linear rationality guided the writing of the narrative. It had a defined beginning and end - although it is open to different interpretations. - as is typical of interactive narratives, but the middle could not be planned in a linear manner. Another logic that directed the narrative was to escape from a predefined narrative genre as set in the project scope.

The story ended up being about a person who doesn´t know their whereabouts and finds themselves in a mysterious situation involving a strange stone portal and alternating dimensions. The multi-dimensional aspect came from the two visual outlooks of the game, being a kind of explanation of why each level has a certain style. The portal element rose from the necessity to transport the player from one level to another. It would also serve as a source for the game's collectable, as some scattered portal segments could be found to complete the unfinished portal in the middle of the forest.

The written text ended up more or less satisfying these prerequisites. Nonetheless, it feels a bit dull and unimaginative. The lack of writing experience along with the short amount of time allotted to this task, are determining factors in its quality. To draft a captivating story is a skill on its own, and the role of the narrative designer is wholly justified in story-driven games and studios that can afford this occupation within its structure.

3.4.3.3 Collectible system

The ability to collect items and use this to advance in the game was the first planned game mechanic for this project. The system follows a similar structure as the previous one, having elements scatted on the level that, once interacted with, triggers a UI element to be displayed on screen to show that that item was collected by the player. The system is controlled by a Game Manager script that not only controls the collecting mechanic but also its integration with other mechanics.

This game manager is what allows the monologue system to be integrated in the collectables so when the player interacts with them, not only does it takes the item, but it also displays a text. It also allows one of the most important text prompt moments, in terms of guidance, that is the message that appears right after all collectables are gathered. This message points the player to go to the stone portal element as this is needed to change the level and advance the game.

The interaction would be possible with the press of another button, in this case, the “E” key on the keyboard, as this is a standard key used in games for interaction and activation of game elements that would be familiar to players.

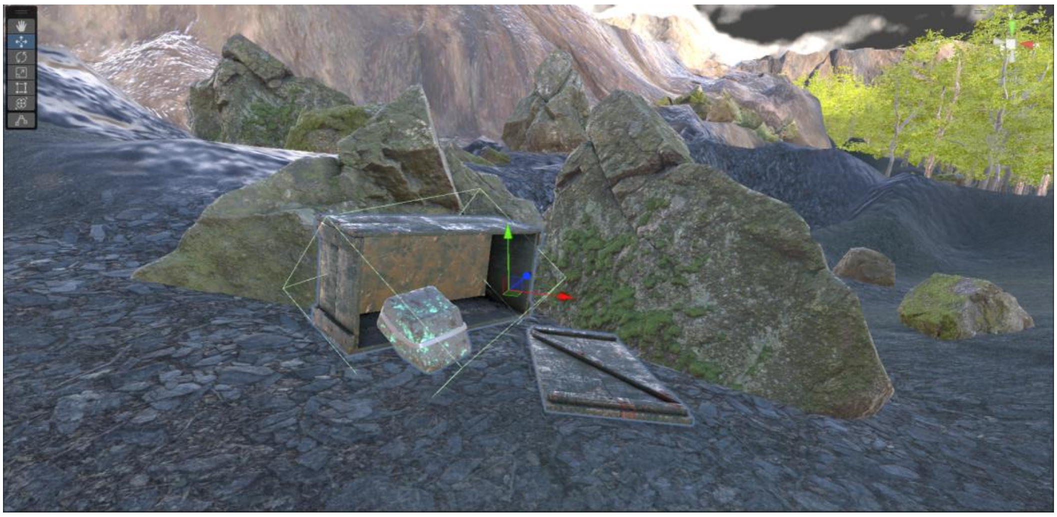

Figure 62 - In this case, the asset was positioned so it could be integrated into the environment.

It introduces a new game element which is the collectable itself. This element is composed of a piece of stone, shaped to form a segment of the portal, and a wooden box. The idea behind the box is to induce the player to create a relationship between the portal and the military that once inhabited the base. As the game does not offer any kind of response to the situation the players are presented with, this could offer a possible explanation.

There were eight different collectibles on each level. The stone had a different form for each, the box may have a lid or not, and the arrangement of these parts changed as well. This added some variation to the element to lessen the feeling of repetition that could have arisen from the task of collecting many items. They also suffered a minor change in the stylized scene as the box element was discarded, and only the stone was used.

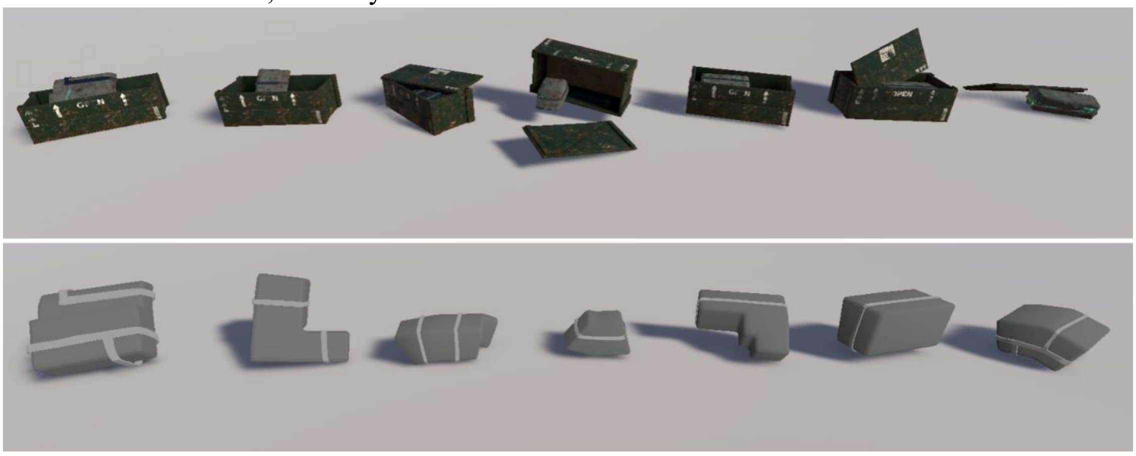

Figure 63 - Each level had seven collectables as seen in both photorealistic and stylized versions.

3.4.3.4 Scene transition

The last mechanic to implement was responsible for the transition between both levels. This would allow the player to go from the photorealistic one to the stylized one and back again. This mechanic was also controlled by the Game Manager as its functioning is dependent on conditions coming from the other mechanics.

Apart from the logic necessary to control the loading transition from one level to the other, this mechanic had a visual aspect to it. To mark the moment in which the player changes visual styles – relevant for the story and the game experience - an animation would be triggered as the player gets close to the portal area and has taken all collectables. The animation was created using Unity’s animation tools. This tool works with a timeline in which keyframes can be set to save various states from a game object aspects. These aspects states interpolate between keyframes creating the animation.

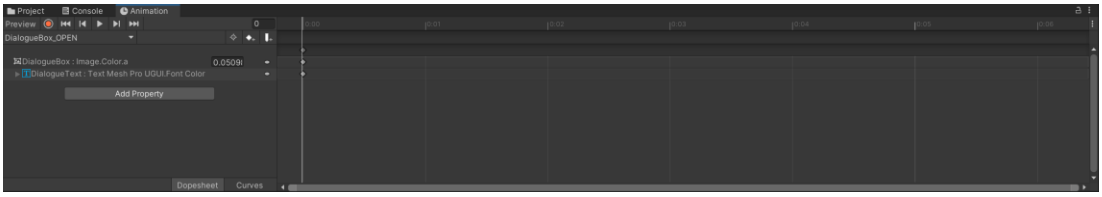

Figure 64 - Unity animation tab allows the control of keyframes in a timeline.

To guarantee that the animation would always be seen in the same position, the game's camera had to be controlled while the player's movement was restricted. This was done whit another script, controlled by the Game Manager, and that makes use of a free package from the asset store called *LeanTween* from Dented Pixel. This package allows the creation of code-driven animations, or tweens, which can be created from inside a C# script. The advantage is that as tween are driven by code, they can solve problems that the animation tool could not. As the direction and exact position from where the player enters the portal area is variable, the code can take this into consideration and adjust the camera position correctly. If this camera movement were done with animation, there would be a gag in its beginnings as the initial position would always be the same, independent from where the player came from.

3.5 Validate/ Conclusion

In a usual Design Sprint session, this phase would be dedicated to the validation of the prototype with users and stakeholders. Usability studies can be done to assess their perception and validate the concept. With the dissertation's major goal being focused on the process rather than the validation of a result, the concerns of the project's development are more relevant to it than the validation of an initial concept.

Therefore, it seems correct to leave this Design Sprint validation process aside and adopt a different metric for the completion of the developed project. Project review processes are often employed at the end of a project development cycle, and one of the most common processes is the use of project post-mortem. In general, it takes the form of a meeting with the project team to reflect on how the project went, and any lessons can be used to modify and develop project practices and procedures (Rasper et al., 2002). It can take many forms, but its structure usually revolves around a comparison of the initial goals with what was actually accomplished, as well as an analysis of why it happened the way it did. It focuses on the process and how to improve it in a next iteration and, by doing so, seems to fit well with the needs of this chapter of the dissertation.

Going back to what were the fundamental intentions of the project, the planned goals were to design and create a digital game prototype that makes use of photorealistic and stylized appearances in a navigable three-dimensional space and that had no defined narrative genre.

Concerning the narrative genre aspect, it is possible to say that the results diverge from the initial plans. The addition of narrative elements to the experience causes the player to unintentionally frame the game into a style category. The very vagueness with which the text was created with the intention of not making it fit into a category causes it to be perceived as "mysterious". Thus, it can fall into the mystery literature category.

This reverberates in the game genre as the mystery elements bleed from the narrative to the game mechanics framing it in the *Mystery Dungeon* genre. Games in this category do not need to have a dungeon *per se* but have a kind of secret to be solved, elements of puzzle and exploration in a nonlinear configuration. The game genre framing is not an issue. On the other hand, it serves as an indicator that the experience offered by the prototype is mature enough to be fitted in a known category.

In terms of spatiality, the game has a fully three-dimensional navigable space in which the player can explore the environment without any kind of constraint, apart from the restrictions related to the physics of the elected method of movement, namely a walking simulation, and the confinement of the game space.

Something noteworthy is how a mechanic so easy to take for granted as this one can have an impact on the feel of the game. To help the sense of photorealism of the first level, the movement had to be tuned to feel realistic too. Setting the simulation parameters to values true to reality, for instance, using $9{,}8m/s^2$ to calculate gravity behavior, did not guarantee that the mechanics felt as realistic. These values had to be adjusted so the movement could have a more realistic feel. Therefore, player experience takes precedence over having the most precise computations and simulations.

Regarding game appearance, results have a major relevance as they are the focus of the dissertation. Overall, the two planned styles appear in the final game, fulfilling one of the project's goals. On the topic of the photorealistic style, the results achieved are not on par with the state-of-the-art games out today, but to compare an AAA game developed by a large team during the course of years would be unfair with a game prototype developed in a much shorter time frame. Thus, it is undeniable that the visual outlook produced was within the style delineations. It made use of the most advanced technology available in the form of Unity’s high-definition render pipeline. It used physically based simulations, especially to create lighting conditions, and all game elements were created to look as photorealistic as possible.

The appearance reached on the stylized level also has a positive result. The cell shading effect was achieved, although the outline effect was not included. The choice of method to make the level viable turned out to be enough to deliver some results within the limitations of the project, but with losses from an artistic point of view. It is preferable to create a visual style from scratch rather than adapting existing elements if possible.

The main lesson learned is how interwoven the artistic and the technical aspects are in a digital game and how important it is to understand this in order to reach the desired outcomes. In future projects, minor technicalities learned from this project will be applied so the development process can run smoothly and with fewer hiccups, as happened this time. Coming from an artistic background, this author feels compelled to get additional technical expertise in order to continue doing artistic jobs while better understanding the requirements and limits that digital game art demands.

It is also worth returning to the list of assumptions presented at the beginning of the project to compare whether the opinion on each point remains the same. This comparison is presented in the form of a table that shows

Table 1 - List of commented project assumptions.

Original statement	Comment
The development of the photorealistic appearance would be, in general, harder than the stylized one.	Although the photorealistic level demanded a large amount of time, in terms of developing a visual outlook, the amount of effort was the same for both appearances.
The work of 3D modeling would be the most arduous and would require more execution time, which could become a bottleneck for the project development.	Modeling was only a part of the asset creation work, which included texturing and the integration with the game engine. Of the three tasks presented, texturing was the most requiring one both in terms of time and level of difficulty.
To create an interesting 3D environment would be a straightforward process, easy to achieve.	To create game levels and environments is not straightforward, in fact, it needs a huge amount of back and forth and it depends on the integration of many elements.
Great support would come from the Unity community on the internet, via blogs, vlogs and forums. Any problem found during development was already solved and its solution was made available online.	There is a huge amount of content online but as the problems became more specialized is hard to find solutions for them. Also, some topics have an entering barrier in the form of prior knowledge one needs to have to understand the message being sent.

4. Conclusion

This dissertation focused on the analysis of digital games' visual styles and how to implement two of the most standard styles in a game prototype. After the problem and research objects were defined, the work began with a literature survey on the topic of digital games' visual style, in which the elements that define the appearance of a game were indicated so a ground base could be formed to support the discussion. Also, photorealism in CGI and digital games were explored to better understand its definitions and limitations as a visual style. The stylized visual was also explored with the same intention as both visual styles would be used in a practical exploration in the next section.

Using the Design Sprint methodology to structure its creation, a digital game prototype was created to help understand the difficulties of incorporating the photorealistic and stylized look into a game, in artist and technical terms.

The study was done but not without some limitations. In practical terms, the prototype project had to stay inside its boundaries of scope, time, and cost, which drove its overall quality. Being this a scenario of development by a single person, the competency of this individual has a massive influence on the final outcome. The experience had during the project would vary if someone with a different skill set had done it. That said, it is hoped that some of the lessons learned in the process can be extrapolated and somewhat generalized to be helpful for novice designers and developers wishing to enter the digital game independent industry.

4.1 Objective Fulfillment

The objective of the dissertation was to understand the defining parameters of a digital game visual style by defining, firstly, what it is and how two of its most prominent variations are characterized. Secondly, visual style was explored in a more practical scenario through the development of a project aiming to create a game prototype so that visual style would be understood theoretically, and the firsthand experience would provide a practical facet to the study.

The combined results of the two components of the study were put into a comprehensive guideline for artists, designers, and developers who would like to start working in independent game production. The focus is on visual style for games, but some of the topics can touch on other points such as game design or level design, as sometimes these fields can intersect each other.

This guideline is not supposed to be an exhaustive list of dos and don’ts, but it synthesizes our experience after the completion of the project. The list is presented in the table below.

Table 2 - Guidelines for visual style implementation for independent game development.

Game Art is a complex and delicate subject that should not be taken for granted.
• In the same manner as a sculptor who works with clay has to understand how the material behaves, its nuances and qualities to gain more confidence and experience in his practice, the game artist needs to understand their media of choice. With more practice and familiarity in the software ecosystem, one gains better control of the outcome as there are many minute details that can make a significant difference. Acquiring some programming skills is important as this will give better understand of the logic of the system helping with the overall understanding of the media they are working on. • Experimentation and iteration are useful processes to help to achieve the desired results. As much as the creation method may seem linear, in practice, the process has many back and forth moments so new elements can be assessed. If the test fails, as it will definitely happen, it is a matter of going some steps back, iterating on the concept and testing again. It is useful to create a *playground* scene in Unity to evaluate every new game element prior to implementing it in the correct scene.
The photorealistic look emerges from the sum of every detail on the screen.
• To achieve an acceptable photorealistic look is to focus on the minutiae. The world is full of imperfections, and a considerable part of the sense of realism will come from the flaws introduced to the game elements. Glass always has some grease or dust, wood never repeats in a perfect pattern, concrete flakes and form dust around it. The list can go indefinitely. This issue implies adding imperfections to the 3D model’s mesh in the shape of a dent or bump or ever breaking the symmetry of the model by slightly deforming it. More frequently than that, defects can be included in the model's texture set because they add a substantial amount of surface data. It is important to

have deep knowledge of how textures work as they are fundamental to achieving the style.

- Photorealism can be described as a style about fooling the viewer's perception, creating the impression of visual realism. This is easier to achieve when one is not trying to replicate natural reality in itself, but when one understands that photorealism can be achieved by the simulation of natural elements as seen through a physical camera.

  Cameras add distortions to the image that are more familiar than the crisp, pure digital look that a Unity default camera has. To simulate physical camera behavior, in tandem with a correct light simulations, can elevate the sense of photorealism of the project.

Three-dimensional styled graphics require the development of a particular skill set.

- The necessity to have a greater understanding of the technological system becomes more evident when creating a stylized look for a digital game. Replicating a kind of stylization is straightforward because there are many resources on the internet that address the subject, but creating a unique stylization requires a combination of artistic intention and technical expertise.

  Stylization often relies on recreating a traditional artistic technique to the digital media. This adaption is strongly reliant on technicalities to be achieved; hence it is difficult to avoid obtaining the essential technical abilities. Delving into the topic of shaders is advised as it is at the core of this matter.

- As there are many possibilities under the stylized term, it is important to define the type of stylization prior to the start of the project. It is usually recommended to conduct research and obtain references, but an iterative approach might yield excellent outcomes.

Nonetheless, the ultimate outcome of this dissertation is expected to be satisfactory and useful to not only academics but, as previously indicated, practitioners. It achieved its purpose of expanding understanding of these themes and raising awareness of the relevance of visual style in digital games.

4.2 Future Works

This dissertation focused on the design and development side of visual outlooks in digital games. The player side and their impressions are yet to be explored. A survey to better understand gamer visual style preferences and if there is a connection between visual style and player motivation profile so studios and game designers can choose a style with research grounded data.

Another possible unfolding is to continue the prototype concept to create a full game. The game concept was validated in the experiment together with the technicalities of the game engine, demonstrating that it could evolve into a proper game. The use of two different visual styles is something underexplored in games and could be a differentiation point for this one.

The development of the prototype was a great experience that showed promising professional possibilities for the future as well. Individual game development forces one to wear many hats and, with that, to try different occupations and assume multiple responsibilities within the development framework. Among the various roles, the career of technical artist stood out as an attractive professional path to pursue, and some effort will be made in that direction.

5. References

Aarseth, E. (2001). Allegories of Space. The Question of Spatiality in Computer Game. In *Zeitschrift fur Semiotik* (pp. 152–171).

AL82 Retrogaming Longplays. (2014). *Teenage Mutant Ninja Turtles: The Arcade Game Longplay (Arcade) [60 FPS]*. https://www.youtube.com/watch?v=2jLO1upcd8w

Asana. (2021). What is the project management triangle and how can it help your team? *Asana*. https://asana.com/resources/project-management-triangle

Aubrey, A. (2013). Casual Games, Time Management, and the Work ofAffect. *Ada: A Journal of Gender, New Media, and Technology*, *2*. https://doi.org/10.7264/N3ZW1HVD

Beekman, C. (2014, November 17). *The History Of Video Games And The Military*. https://taskandpurpose.com/entertainment/us-militarys-close-history-video-games/

Bordwell, D. (1998). *On the History of Film Style*. Harvard University Press.

Cook, M. (2018). Toy Story at 25: How Pixar's debut evolved tradition rather than abandoning it. *The Conversation*. https://theconversation.com/toy-story-at-25-how-pixars-debut-evolved-tradition-rather-than-abandoning-it-149873

Creative Shrimp. (2021, April). *What is Stylized 3D Graphics and Non-photoreal (NPR)?* Youtube. https://www.youtube.com/watch?v=UchOWwcwm_s

CuriousMarc. (2017). *Lyle Bickley explains the PDP-1 (and we play the original Spacewar!)*. https://www.youtube.com/watch?v=1EWQYAfuMYw

Darley, A. (2002). *Visual Digital Culture* (0 ed.). Routledge. https://doi.org/10.4324/9780203135204

Digital Equipment Corporation. (1960). *PDP-1 Handbook*. http://www.bitsavers.org/pdf/dec/pdp1/F15D_PDP1_Handbook_Oct63.pdf

Epic Games. (n.d.). *Download Unreal Engine: Licensing Options*. Retrieved January 12, 2022, from https://www.unrealengine.com/en-US/download

Fabito, B., & Cabredo, R. (2019). *Exploring Game Aesthetics as Antecedents of Game Continuance: An Analysis in the Lens of Self-Determination Theory*. 7.

Fiadotau, M. (2018). Indie Game. In N. Lee (Ed.), *Encyclopedia of Computer Graphics and Games* (pp. 1–4). Springer International Publishing. https://doi.org/10.1007/978-3-319-08234-9_188-1

Finkelstein, A. (2020). *Non-photorealistic Rendering (NPR) [slides]*. Princeton University Computer Science Department. https://www.cs.princeton.edu/courses/archive/spring20/cos426/lectures/Lecture-16.pdf

Fleming, J. (2007). Down the Hyper-Spatial Tube: Spacewar and the Birth of Digital Game Culture. *Game Developer*. https://www.gamedeveloper.com/design/down-the-hyper-spatial-tube-spacewar-and-the-birth-of-digital-game-culture

Frasca, G. (2003). *Ludologists love stories, too: Notes from a debate that never took place*. https://ludology.typepad.com/weblog/articles/frasca_levelUP2003.pdf

Gee, L. L. S., & Dolah, J. (2018). Graphic Styles Appearance: Perception of Malaysian Students. In R. Anwar, M. Mahamood, D. H. Md. Zain, M. K. Abd Aziz, O. H. Hassan, & S. Z. Abidin (Eds.), *Proceedings of the Art and Design International Conference (AnDIC 2016)* (pp. 117–123). Springer Singapore. https://doi.org/10.1007/978-981-13-0487-3_14

Gersh-Nesic, B. (2019). Origins and Schools of Abstract Art: Art Without a Subject. *ThoughtCo.* https://www.thoughtco.com/what-is-abstract-art-183186

References

Google. (n.d.). *Design Sprint Methodology*. Retrieved May 20, 2022, from https://designsprintkit.withgoogle.com/methodology/overview

Horban, O., Martych, R., & Maletska, M. (2019). Phenomenon of Videogame Culture in Modern Society. *STUDIA WARMIŃSKIE*, *56*.

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). *MDA: A Formal Approach to Game Design and Game Research*. 5.

Järvinen, A. (2009). *Gran Stylissimo: The Audiovisual Elements and Styles in Computer and Video Games*. 16.

Juul, J. (2001). Games Telling stories? A brief note on games and narratives. *Game Studies. The International Journal of Computer Game Research*. http://gamestudies.org/0101/juul-gts/

Juul, J. (2010). *A Casual Revolution: Reinventing Video Games And Their Players*. The MIT Press.

Juul, J. (2019). *Handmade Pixels Independent Video Games and the Quest for Authenticity*. MIT Press. https://ieeexplore.ieee.org/servlet/opac?bknumber=8951008

Kasavin, G. (2004). Metal Gear Solid 3: Snake Eater Review. *GameSpot*. https://www.gamespot.com/reviews/metal-gear-solid-3-snake-eater-review/1900-6113331/

Kimber, M. (2021). Kena: Bridge of Spirits—Graphics are main selling point for this game and they do not disappoint. *Sky News*.

Langford, M. (2000). *Basic Photography* (7th ed.). Focal Press.

Lei, Z., Xie, Y., Ling, S., Pastor, A., Wang, J., & Callet, P. L. (2021). Multi-Modal Aesthetic Assessment for Mobile Gaming Image. *ArXiv:2101.11700 [Cs]*. http://arxiv.org/abs/2101.11700

Lewis, E. (2004, November 17). *Metal Gear Solid 3: Snake Eater Review*. IGN. https://www.ign.com/articles/2004/11/17/metal-gear-solid-3-snake-eater

Lipkin, N. (2013). Examining Indie's Independence: The Meaning of "Indie" Games, the Politics of Production, and Mainstream Co-optation. *Loading... The Journal of the Canadian Game Studies Association*, *7*, 8–24.

Loveridge, S. (2022). *The Legend of Zelda Breath of the Wild Captured Memories locations*. https://www.gamesradar.com/legend-of-zelda-breath-of-the-wild-captured-memories-locations-guide/

Manovich, L. (2002). *The Language of New Media*. MIT Press.

Marder, L. (2020, January 24). The Definition of "Form" in Art. *ThoughtCo.* https://www.thoughtco.com/definition-of-form-in-art-182437

Marks, T. (2018). *Return of the Obra Dinn* [Video game screenshot]. https://www.ign.com/articles/2018/10/22/return-of-the-obra-dinn-review

McCloud, S. (1993). *Understanding Comics*. Kitchen Sink Press.

McDonnell, R., Breidt, M., & Bülthoff, H. H. (2012). Render me real?: Investigating the effect of render style on the perception of animated virtual humans. *ACM Transactions on Graphics*, *31*(4), 1–11. https://doi.org/10.1145/2185520.2185587

McLaughlin, T., Smith, D., & Brown, I. (2010). A framework for evidence based visual style development for serious games. *FDG 2010 - Proceedings of the 5th International Conference on the Foundations of Digital Games*, 132–138. https://doi.org/10.1145/1822348.1822366

Mori, M., Kageki, N., & MacDorman, K. (2012). The Uncanny Valley: The Original Essay by Masahiro Mori. *IEEE Robotics & Automation Magazine.*, *19*(2), 98–100. https://doi.org/10.1109/MRA.2012.2192811

Nakamae, E., & Tadamura, K. (1995). Photorealism in computer graphics—Past and present. *Computers & Graphics*, *19*(1), 119–130. https://doi.org/10.1016/0097-8493(94)00127-K

Nake, F. (2018). Someone Is Always Already There In Front Of You Even Though You May Not Like It. *Journal of Science and Technology of the Arts*, *10*(3), 11.

References

Nitsche, M. (2008). *Video game spaces: Image, play, and structure in 3D game worlds*. MIT Press.

Old Classic Retro Gaming. (2014). *Arcade Game: Atari Football (1978 Atari)*. https://www.youtube.com/watch?v=YGDzW6JCGMs

Paramount Pictures. (2021, June 29). *Clifford the Big Red Dog (2021)—Official Trailer—Paramount Pictures*. https://www.youtube.com/watch?v=4zH5iYM4wJo

Peddie, J. (2013). *The History of Visual Magic in Computers—How Beautiful Images are Made in CAD, 3D, VR and AR*. Springer.

Pharr, M., Jakob, W., & Humphreys, G. (2004). *Physically Based Rendering: From Theory To Implementation*. https://www.pbr-book.org/

Pietrak, A. (n.d.). *Nintendo Arcade Game, Restored* [Photography]. Retrieved June 15, 2022, from http://collectedit.com/shops/toomucharcadestuff/nintendo-donkey-kong-upright-arcade-game-restored-very-nice-original-classic-60

Pluralsight. (2015, September 15). Game jobs and what they do. *Pluralsight*. https://www.pluralsight.com/blog/film-games/game-jobs

Rasper, V., Stainer, M., & Carluccio, P. (2002). A real-life approach to lessons learned. *Project Management Institute Annual Seminars & Symposium*. https://www.pmi.org/learning/library/real-life-approach-lessons-learned-8497

Ribbens, W. (2013). Perceived Game Realism: A Test of Three Alternative Models. *Cyberpsychology, Behavior, and Social Networking*, 31–36. https://doi.org/10.1089/cyber.2012.0212

Riot Games. (2018a). *So You Wanna Make Games?? | Episode 2: Concept Art* [Youtube video]. https://www.youtube.com/watch?v=FqX-UMVTLHI&list=PL42m9XiTqPHJdJuVXO6Vf5ta5D07peiVx&index=3

Riot Games. (2018b). *So You Wanna Make Games?? | Episode 5: Technical Art* [Youtube video]. https://www.youtube.com/watch?v=kr7XYXMM7-U

Ryan, M.-L. (2001). Beyond Myth and Metaphor – the case of narrative in digital media. *Game Studies. The International Journal of Computer Game Research*. http://gamestudies.org/0101/ryan/

Schell, J. (2015). *The Art of Game Design: A Book of Lenses* (2nd ed.). CRC Press.

Schreier, J. (2021). Sony's Breakout Video Game Owes its Success to a Hazmat Suit. *Bloomberg*. https://www.bloomberg.com/news/newsletters/2021-10-22/sony-s-hit-video-game-for-the-playstation5-almost-never-happened-in-covid

Sheff, D. (1999). *Game Over Press Start To Continue* (1st Edition). Cyberactive Media Group Inc/Game. https://archive.org/details/0966961706/page/n3/mode/2up

SourceSpy91. (2016). *Metal Gear Solid 2—Normal Difficulty Walkthrough—No Commentary*. https://www.youtube.com/watch?v=2qsHbwpvHcA

Strategy First. (2005). Strategy First to Deliver Multiple Titles On-Line via Steam [Company website]. *Strategy First*. https://web.archive.org/web/20060328053420/http://www.strategyfirst.com/press/DisplayArticle.asp?sLanguageCode=EN&iArticleID=3250

Swink, S. (2008). *Game Feel: A Game Designer's Guide to Virtual Sensation*. Morgan Kaufmann.

Unity Technologies. (2022). *Unity User Manual 2021.3 (LTS)*. Unity Technologies. https://docs.unity3d.com/Manual/index.html

Valve Corporation. (2021). *Best of 2021*. https://steamcommunity.com/games/593110/announcements/detail/5071614219310754127

Wang, F., & Hannafin, M. (2005). Design-Based Research and Technology-Enhanced Learning Environments. *ETR&D*, *53*(No. 4), 5–23.

Weisstein, E. (n.d.). Dimension. *MathWorld--A Wolfram Web Resource*. https://mathworld.wolfram.com/Dimension.html

References

Whalen, A. (2019). The Illusion and emotion Behind “Toy Story 4.” *Newsweek*. https://www.newsweek.com/toy-story-4-animation-bts-behind-scene-1447615

Wolf, M. J. P. (2001). *The Medium of the Video Game* (1st ed.). University of Texas Press.

Wolf, M. J. P. (2013). *Abstraction in the Video Game*. 10.

Zell, E., Aliaga, C., Jarabo, A., Zibrek, K., Gutierrez, D., McDonnell, R., & Botsch, M. (2015). To stylize or not to stylize? The effect of shape and material stylization on the perception of computer-generated faces. *ACM Transactions on Graphics*, *34*(6), 184:1-184:12. https://doi.org/10.1145/2816795.2818126

Zgeb, B. (2021). Pixelate filter: Post-processing in a compute shader [Blog]. *Bronson Zgeb*. https://bronsonzgeb.com/index.php/2021/07/17/pixelate-filter-post-processing-in-a-compute-shader/

6. Ludography

Atari (1972). Pong. [Arcade], Atari Japan.
Atari (1976). Football. [Arcade], Atari.
BioWare (2017). Mass Effect: Andromeda. [Microsoft Windows, PlayStation 4, Xbox One], Electronic Arts.
Broken Rules. (2009). And Yet It Moves. [Microsoft Windows, Mac OS X, Linux], Broken Rules.
DMA Design Limited (2001). Grand Theft Auto III. [PlayStation 2], Rockstar Games.
Ember Lab. (2021). Kena: Bridge of Spirits. [Microsoft Windows, PlayStation 4, PlayStation 5], Ember Lab.
Japan Studio, Team Ico (2001). Ico. [PlayStation 2], Sony Computer Entertainment.
Konami (1989). Teenage Mutant Ninja Turtles. [Arcade], Konami
Konami Computer Entertainment Japan (2005). Metal Gear Solid II. [PlayStation 2], Konami.
Nintendo Entertainment Planning & Development (2017). Mario Odyssey. [Nintendo Switch], Nintendo.
Nintendo Entertainment Planning & Development (2017). The Legend of Zelda: Breath of the Wild. [Nintendo Switch], Nintendo.
Number None (2008). Braid. [Xbox 360], Number None, Microsoft Game Studios.
Pearl Abyss (2022). DokeV. Pearl Abyss.
Pope, Lucas (2018). Return of Obra Dinn. [Microsoft Windows, Mac OS X], Studio 3909.
Russell, Steve (1961). Spacewar. [DEC PDP-1]
Smilebit (2002). Jet Set Radio Future. [XBox], Sega.
Square (1997). Final Fantasy VII. [PlayStation], Square Enix, Sony Computer Entertainment.
Studio Pixel (2004). Cave Story. [Microsoft Windows], Studio Pixel.
Team Silent (2001). Silent Hill 2. [PlayStation 2], Konami.

Appendices

Appendix A: Monologue Script

Photorealistic Scene

Trigger A (starting position)
…where am I? It feels like I banged my head against a brick wall.

Trigger B (cabin door)
This door is closed shut.
I think I hear someone inside but maybe is just these squeaking floorboards.

Trigger C (collectable A)
There is a strange stone inside this box. Looks important. I feel I should take it with me.

Trigger D (trail cabin-road)
There is a road up ahead. I would say That I came from this direction but I have no memory of it.
I don't see any signs of a car and if I came from the road I should have used a car, right?
Maybe bus?

Trigger E (road-falling tree)
The tree is blocking the way. I can´t continue in this direction.

Trigger F (road gasStation)
A gas station? Maybe I'll finally find a living soul in this place.
And if I do find someone, it would be great if they could give me some information about what is happening right now.

Trigger G (gas store front)

The reflective film on the windows does not let me see inside and the door sensor doesn't seem to be working.

I can't hear anyone inside. Aren't gas stations supposed to be open 24/7?

Can I start to feel frustrated now?

Trigger H (collectable B)

Another one of those box with a stone inside.

Trigger I (road reasoning 01)

I feel like I should remember this place somehow. It seems like I've passed this chunk of the road many times but there is so little here that I registered this place as "unimportant". I still have no idea how or why I've ended up here of all places.

Trigger J (jersey barriers)

Humm, this side is blocked too.

This seems to explain the lack of road traffic but there is still no explanation to why is it blocked in the first place.

Trigger K (clearing)

This is a strange thing to find in the side of a random road.

Never seen anything like this before but it seems to relate with those rocks I keep discovering.

Trigger L (front of Military Base)

This place looks abandoned…

Maybe I still can find some information about what is happening inside this place.

Trigger M (Watchtower)

The view from this height would be nice if it weren't this thick morning fog.

Trigger N (collectable E - watchtower)

Another stone in a crate.

Why would someone put it here? Were they hiding it for some reason?

Trigger O (building entrance)

This place is in a precarious state. I should be careful not to disrupt a structural piece of concrete.

Trigger P (building interior)

What am I even doing in a place like this?

Trigger Q (collectable C)

I hope these stones help in any way. I should take this one too.

Trigger R (near tents)

This clearly is a military facility but I don't remember having any connection with the military.

I feel as I'm trespassing but in the situation I'm in right now if I should do anything different than explore it.

Is it considered transgression to trespass a military base?

Trigger S (collectable F)

Maybe someone was trying to take this one to the helipad?

Trigger T (collectable D)

Another rock! They sure are a lot lighter than they look.

My back thanks for that. That is for sure.

Trigger U (collectable G)

No sight of the box. Only the lid remains.

And the stone, of course. I have to take it.

Trigger V (trail reasoning 01)

This trail seems to be used a lot. Its wide enough for a small vehicle but I don't see tire marks anywhere.

I don't see any footprints too so there is that.

- When all collectables are gathered

I think I have all the parts to complete that stone thing in the clearing.

My gut tells me that I should head back there to see if I can stack these ones with the ones over there.

Stylized Scene

Trigger A (starting position)
I'm back at the cabin. The headache is also back.
This time my eyesight seems to be affected as well. Or is not my eyesight?
What the heck is happening here?!

Trigger B (trail cabin left)
This trail was here before. I took me to that military facility.
These trees look weird. I'm not sure if is really a sight problem…

Trigger C (trail end)
The other time there was one of those stones here. Now there is nothing...

Trigger D (collectable A)
Ok, found a stone. Let´s take it.

Trigger E (trail reasoning)
Everything is different but the same.
Even the air seems different than before.

Trigger F (front military)
The military base continues in the same place as before. The collapsing building and the tents looks the same.
Let´s check it out.

Trigger G (military building entrance)
I don't want to enter this place again.

Trigger H (collectable C)
I wonder where these stones came from…

Trigger I (building later collectable place)
Nothing here.
It seems I got up here for nothing…

Trigger J (watchtower)
It's definitely not my eyes! The world changed!
What the heck is going on? Where am I?!

Trigger K (collectable F)

If this is another world are the stones from this place or the other, originally?

Trigger L (collectable G)

I wonder if I'm dealing with a magic related situation or a technology based one.

What am I saying? This seems like my brain is already trying to rationalize this mad experience.

Trigger M (portal)

This damn portal is in the center of this all.

I need to rebuild it one more time to understand what is happening right now!

Trigger N (collectable E)

How is it possible that I've stack these and now they appear in random places?

Trigger O (trail cabin right)

I have no sense of time. I wonder how long my blackout lasted.

Maybe those trees were removed and the road in open now.

Trigger P (block path - trees)

Road still blocked.

Damn!

Trigger Q (road reasoning)

I feel I was walking in this road a moment ago, but it sure feels different, at the same time.

I can't say if this is more familiar than the other one….

Trigger R (gas station)

The gas station seems as deserted as before. Not a single soul to help me figure this out.

Trigger S (collectable B)

I guess I'll have to keep taking these with me.

Trigger T (collectable D)

This sure looks like made of stone and metal but the lack of weight still fascinates me.

These are from another world!

Wait…

Trigger U (road reasoning 02)

Is this an alternate reality or was I in the alternate version before?

Both seem as strange and familiar as the other. I feel I've passed through this road many times but can't remember much more than this.

Trigger V (block road - blocks)

Still block by these elements.

Something wants me to stay here…

- When all collectables are gathered

Ok. I got all stones again. Now its time to go to the bottom of this!

I should go back to that clearing where the stone foundation is and assemble the structure one more time.

Appendix B: Gameplay Recording

A recording of the gameplay can be found at the link below:

https://www.youtube.com/watch?v=V2-Ak_6ADJQ