TECHNISCHE HOCHSCHULE LÜBECK

Bachelor thesis

Anxiety Disorder and its Representation in Video Games

Submitted by: Clara Bohnsack

Department: Electrical Engineering and Computer Science
Degree program: Information Technology and Design
First examiner: Fabio Anthony

Date handing out: 1st June 2024
Date handing in: 28st June 2024

Task description

Over the last few decades, Mental Health has been covered significantly more in media and conversation. Hereby, the portrayal of such illnesses in Video Games has ranged from being the main driving force of the game, as is the case in the title *Hellblade: Senua's Sacrifice*, to the topic being more subtly implemented in games like *Baldur's Gate 3*. There are plenty of other examples of mental illness depictions in games, however, few of these accurately display the underlying workings. This paper attempts to add to the conversation by discussing how the portrayal of symptoms of particularly anxiety disorder could be more accurately displayed and merged into a coherent gameplay loop. A prototypical application is then created based on the findings.
The task for this work is:

1. Create an Action-Adventure prototype that playfully highlights anxiety disorder symptoms
2. Create an interactive Windows-App using the Unreal Engine 5 (UE5)
3. Using a First-Person camera perspective to maximize immersion
4. Create one playable level
5. Implement a graphical style that is inspired by the works of artist Moebius
6. Implement a storyline that weaves various characters together through dialogue
7. Create visual and audio cues that represent a depiction of anxiety symptoms

Optional features are:

1. Valuate to what extent the effects of anxiety symptoms can have a direct influence on a game's combat system and environment
2. Create a release candidate of the prototype for the Windows (10+) operating system

TECHNISCHE HOCHSCHULE LÜBECK

Eigenständigkeitserklärung

Declaration of Originality

Bohnsack, Clara

Name, Vorname
Last name, first name

324278

Matrikelnummer
Matriculation number

Ich versichere hiermit, dass ich die vorliegende
I hereby declare that this

☐ **Hausarbeit** term paper ☒ **Bachelorarbeit** bachelor's thesis ☐ **Masterarbeit** master's thesis

mit dem Titel
with the title

Anxiety Disorder and its representation in Video Games

eigenständig und ohne unerlaubte fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und Entlehnungen aus anderen Arbeiten kenntlich gemacht. Für den Fall, dass die Arbeit zusätzlich elektronisch und/ oder digital eingereicht wird, erkläre ich, dass die schriftliche und die elektronische und/ oder digitale Form identisch sind. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

is my own original work and any assistance from third parties has been acknowledged. I have clearly indicated and acknowledged all sources and resources as well as any borrowings from other works. In case of an additional electronic and/or digital submission of this work, I declare that the written form and the electronic and/or digital form are identical. This work has not previously been submitted either in the same or in a similar form to another examination office.

Ich bin damit einverstanden, dass die vorliegende Hausarbeit/ Bachelorarbeit/ Masterarbeit für Veröffentlichungen, Ausstellungen und Wettbewerbe des Fachbereiches verwendet und Dritten zur Einsichtnahme vorgelegt werden kann.

I agree that this work can be used for publishing, exhibition or competition purposes and can be inspected by third parties.

☒ **ja** Yes ☐ **nein** no ☐ **es liegt ein Sperrvermerk bis _____vor** there is an embargo period until

Sülfeld, 25.06.2024

Ort, Datum
Place, Date

Unterschrift
Signature

Abstract of Thesis

Fachbereich: **Department:**	Electrical Engineering and Computer Science
Studiengang: **University course:**	Information Technology and Design
Thema: **Subject:**	Anxiety Disorder and its Representation in Video Games
Zusammenfassung: **Abstract:**	In recent years, the topic of mental illness has been increasingly picked up in the media. However, representation in media is still often burdened by stigma and passes on prejudices that tend to harm those affected in the long term. Therefore, an accurate and differentiated representation of such diseases is relevant to counteract this issue. One specific medium in which the representation of individual mental illnesses has already taken place is the medium of video games. This thesis aims to take up the topic of mental illnesses, look at and evaluate existing examples, and create a relevant representation of a generalized anxiety disorder using a game prototype developed in Unreal Engine 5 and thus contribute to the discourse. Symptoms of a generalized anxiety disorder are recreated audio-visually and established as playable skills. During implementation, care is taken not to romanticize the illness but at the same time not to catastrophize it.
Verfasser: **Author:**	Clara Bohnsack
Betreuender Professor: **Attending professor:**	Fabio Anthony
WS / SS:	SS 2024

Zusammenfassung der Arbeit

Fachbereich: **Department:**	Elektrotechnik und Informatik
Studiengang: **University course:**	Informationstechnologie und Design
Thema: **Subject:**	Anxiety Disorder and its Representation in Video Games
Zusammenfassung: **Abstract:**	Über die letzten Jahre sind psychische Krankheiten immer öfter medial aufgegriffen worden und eine allgemeine Aufklärung über sie hat stattgefunden. Jedoch ist die Repräsentation in den Medien noch oft belastet von Stigmata und leitet Vorurteile weiter, die den Betroffenen langfristig schaden. Eine akkurate und differenzierte Repräsentation solcher Krankheiten ist somit relevant, um Vorurteile zu bekämpfen. Ein spezifisches Medium, in dem die Darstellung von einzelnen psychischen Krankheiten bereits stattgefunden hat, ist das Videospiel. Gegenstand dieser Arbeit ist, das Thema der psychischen Krankheiten aufzugreifen, bestehende Beispiele zu betrachten und zu bewerten und anhand eines in der Unreal Engine 5 entwickelten Spiel-Prototypen eine relevante Repräsentation einer generalisierten Angststörung zu schaffen und damit zum Diskurs beizusteuern. Dabei werden Symptome einer generalisierten Angstörung audiovisuell nachempfunden und als spielerische Fähigkeiten etabliert. Bei der Implementierung wird darauf geachtet, die Krankheit nicht zu romantisieren, aber im gleichen Zuge nicht zu katastrophalisieren.
Verfasser: **Author:**	Clara Bohnsack
Betreuender Professor: **Attending professor:**	Fabio Anthony
WS / SS:	SS 2024

Contents

1 Introduction

In the last years, the topic of Mental Health and Mental Illness has been discussed in mass media and pop culture increasingly. While the speed of interest grew, the topic is still heavily stigmatized [1]. Common misconceptions are seeing people with Mental Illness as dangerous and irrational and painting Mental Illness as something hopeless and incurable [2][3]. On the other hand, Mental Illness is increasingly often romanticized by younger generations, with Social Media playing a significant role [4].

> "The allure of negativity, packaged in aesthetically pleasing edits and shared experiences, captivates young audiences, blurring the lines between universal emotions and genuine mental health symptoms." [4]

The role of an accurate and balanced depiction of various mental illnesses plays an integral part in dissolving said stigmata. This thesis aims to contribute to the conversation and stray away from depicting mental illnesses as dangerous and further into a positive approach, which is nuanced enough not to lean towards romanticism.
The thesis and the game accompanying it attempt to connect to the eudaimonic approach to well-being, which focuses on self-fulfillment, meaning, and growth, rather than the hedonic approach, which centers fulfillment around fulfilling base wants and needs. The eudaimonic perspective will materialize by looking inward and finding purpose for oneself. It is maintained by the pursuit of inner growth and authenticity [5].

This thesis aims to create a game that picks up specific mental health topics elaborated in the following chapters. The game *Eye of the Storm* strives to use standard game mechanics and apply the eudaimonic principles to it, resulting in a relatively new approach to the topic. Throughout the game, mental illness must not be catastrophized nor painted in a romanticized way.

1.1 Related works

The topic of Mental Illness and its representation is displayed in several types of media in some form. Some works should be mentioned as they contributed to the conception of this thesis or are adjacent to it in terms of topic or style.

1.1.1 Scientific contributions

Numerous research approaches have been taken concerning the potential of using interactive media to decrease the stigma surrounding Mental Illness. In this section, some of these instances will be explored.
In a paper by Majid Zare-Bidaki et al. [6], the effectiveness of a virtual reality simulation psychosis as a means to increase knowledge and empathy in medical students towards patients with psychosis is tested. During the experiment, one group of students experienced one session of the virtual reality application, and the other visited patients under supervision. Results show effectiveness in reducing stigma and increasing patient knowledge and empathy. Compared to the latter approach, implementing the virtual reality simulation showed more substantial results in reducing stigma.
In [7], Ferrari et al. state that in 97 out of 100 games they reviewed, Mental Illness is portrayed in a harmful, misleading, and problematic way. Mental illness is often displayed as something mysterious, unpredictable, and obscure and as an ongoing struggle or an endless battle.
In [8], a study by Jozef Buday et al. provides insights into the depiction of Mental Illness in Video Games over the last 20 years. Within this selection, approximately 1 out of 10 games portray themes of mental illness, with 75% of these painting the illness in a negative light. Displaying mental illness as something to be scared of or as something hopeless can lead to dangerous stigmata and assumptions about people struggling with these illnesses in real life. The referenced findings suggest that video games have the potential to reduce stigma for mental illnesses. At the same time, most video games do not contribute to the discussion or even influence it negatively.

1.1.2 Video games

There have been many attempts to paint mental illness in a more understanding, accurate way, with varying reception [7]. It is essential to note that a video game can never mirror the experience of living with a mental illness fully, as the player will be able to stop at any given time. In contrast, the illness will continuously progress for people suffering from it. Still, implementing medium-specific traits can make an approximation [9]. Traits important to the game accompanying this thesis include audiovisual style, narration, controls, characters, and the game world.

Hellblade: Senua‘s Sacrifice

Hellblade: Senua‘s Sacrifice is an action-adventure game in third-person that was released on August 8, 2017, by the game studio Ninja Theory [10] [11]. In terms of style, the game was inspired heavily by Nordic mythology. Senua is a warrior from a Pict tribe in Scotland, and

the story takes place around 900 AD. The story begins with Senua embarking on a journey to Hel, the land of the dead, to resurrect her dead lover.

Throughout the game, she experiences auditory and visual hallucinations, which indicate a psychosis. The noticeable audiovisual phenomena are called “darkness" by Senua and are a physical representation of her mental illness. While disruptive, most of the time, these symptoms, in terms of gameplay, are often displayed as helpful, as they will be the key to some of the puzzles the player has to solve throughout the game. A recurring puzzle is the unlocking of doors by finding runes. These runes are depicted close to the door but can only be discerned from a certain angle, prompting the player to move to higher or lower grounds, for instance. These puzzles are a direct attempt at displaying a symptom that people who have psychosis often report, which is the perception of patterns and repetitions in the real world. Senua's auditory hallucinations also act in an ambivalent manner, as they sometimes mock and taunt her but other times give her clues about where to go or how to proceed. When the voices are most disruptive, they judge and mock Senua, talk to each other, and generally result in a loud and confusing soundscape. When they act as a guide, they appear more focused and like an entity [12]. At times, Senua's hallucinations also display calming scenes with vivid colors, adding to the attempt at displaying the duality of the mental illness. Senua's healing journey is not presented as linear, and she has to overcome challenges to improve. When Senua dies, the game will restart at a very recent point, but a large black spot, called the dark rot, on her arm will grow each time, visualizing her mental illness growing [9].

In order to display Senua's psychosis in an accurate way that does not further the stigmata of the illness, the team of Ninja Theory consulted several mental health experts of different specializations [9]. Paul Fletcher, a psychiatrist at the University of Oxford, was one of the primary sources of information concerning the topic of psychosis and its symptoms. Further, Ninja Theory worked closely with Charles Fernythough of the University of Durham. Durham is an expert on voice-hearing, which describes the auditory hallucinations people who have psychosis often experience. Based on his information, the team of Ninja Theory put together the voices Senua hears throughout the game. Furthermore, *Wellcome Trust*, a non-profit organization that funds biomedical research, joined the project to support it financially and by introducing it to focus groups of people living with psychosis. During their meetings, the people affected would describe the visual manifestations of psychosis, and Ninja Theory would later attempt to realize those in the game, as seen in Figure 1.1. For instance, one interviewed individual described the following symptoms:

> “Everything is in bits. Like a photograph that[‘s] torn and put together again. If you move, it‘s frightening."

A final presentation of the state of the game concluded with positive feedback from the Mental Health professionals who acted as consultants to the studio. In their words, Ninja

Theory could successfully implement the personal experiences of people affected by psychosis [13]. The accurate depiction of psychosis, visualized in a way that is understandable even for people not suffering from the illness, allows players to immerse themselves in Senua's person and empathize with her struggles.
Hellblade: Senua's Sacrifice is the primary source of inspiration for the game accompanying this thesis. While this thesis thematizes anxiety disorders, the way *Hellblade: Senuas's sacrifice* visualizes the symptoms of psychosis through audiovisual style, narration, game world, and characters can be translated to this thesis. The game shines a more sympathetic light on Mental Illness, as Senua is presented as strong, brave, and capable of improving her ability to cope with her symptoms [12]. This thesis attempts to achieve this representation as well.

Figure 1.1: In-game footage of *Hellblade: Senua's Sacrifice* [11]

What Remains of Edith Finch

What Remains of Edith Finch was first released on April 25, 2017, by the studio Giant Sparrow and published by Annapurna Interactive [14]. The game is in first-person and falls in the adventure genre. Central to the plot is a girl called Edith Finch, whose journal the player interacts with throughout the game. The whole Finch family died under various, mysterious circumstances, about which the player will continuously learn more.
During the game, the player explores their old family estate and uncovers previously unknown family secrets. In dedicated game segments, the player can learn more about the family members' respective lives and how they eventually died. These segments are realized through different types of mini-games that thematically match the death of the family member, like in one segment where the player controls the swing set from which a young boy named Calvin falls to death. Although having strong undertones of the topic, *What Remains of Edith Finch* does not exclusively focus on mental illness, as the focus lies more so on the family's grief.

The game is mentioned in this section of the thesis because of one segment in particular, in which the death of Lewis Finch is displayed. The story segment starts when the main character reads a letter written by Lewis' therapist to his mother after his death. The letter turns into a narration when the mini-game starts. It explains that Lewis, recently sober after a history of substance abuse, noticed the monotony of his life and found himself withdrawing from real life more often by creating a false world in his head. The false world started as a rudimentary labyrinth but slowly transitioned into him building a settlement, later conquering cities, and arriving at his castle. The mini-game, seen in figure 1.2, consists of two screens: one where Lewis chops fish at his job at a cannery and one where Lewis's false reality is depicted. The latter starts relatively small and two-dimensional but slowly increases in size, and its content gets more detailed when Lewis arrives at the settlement, eventually turning three-dimensional and taking over the entirety of the screen when he arrives in his castle. The player controls both screens separately, with the one representing Lewis becoming more challenging to control as it gets pushed to the background. The segment ends with Lewis getting coronated at his castle. He moves towards his queen, holding the crown, and bends down into a guillotine. The screen turns black as a loud cutting noise is heard, which symbolizes his suicide, as can be gathered from the narration [9] [15].
This segment of *What Remains of Edith Finch* is relevant to mention, as it incorporates the controls of a game into the depiction of mental illness. The player controls both Lewis's reality and fiction, with fiction taking over and making it harder for him to act in reality [9].

Figure 1.2: In-game footage of What Remains of Edith Finch [16]

Gris

Gris is a Jump-'n'-Run action-adventure game first released on December 13, 2018. It was developed by the studio *Nomada Studio* and published by *Devolver Digital* [17]. The game follows a girl named Gris who lost her voice. Throughout the game, she goes through different stages that seem to represent the stages of grief. The first stage starts out white, like a blank canvas, but quickly gains reds, blues, greens, and yellows after completing each level. The game is about Gris coming to terms with her complex emotions and does so in an eudaimonic way. The game is mentioned because of its unique approach to providing visual feedback on Gris' journey through colors, as a similar quality is implemented in the game this thesis is based on. This is detailed in 3.2.2 and 3.7 [18].

Night in the Woods

Night in the Woods is a two-dimensional adventure game released on February 21, 2017 [19]. The game is a dialogue-heavy side-scroller. At the center of the story is a woman named Mae Borowski, who dropped out of college prior. She returns to the town she grew up in and finds out how much has changed. After the town's mine closed, it spiraled into an economic downfall. Throughout the game, she reconnects with her old friends and confronts her feelings and shortcomings. Due to Mae's newfound free time, the player can roam around the town for most of the game but will still be driven by significant plot points. After playing for some time, the state of Mae's Mental Health gets picked up increasingly, with heavy implications that she suffers from depression, anxiety, and depersonalization or derealization symptoms, meaning she feels detached from her reality. A more in-depth definition for both depersonalization and derealization as symptoms can be found in chapter 2.2.1. While the game tackles these heavy topics, it does so while being uplifting and optimistic [20] [21].

The reason *Night in the Woods* is mentioned in this chapter is its eudaimonic approach. The game satisfies the player, but not solely due to its gameplay. It does so by attempting to give reassurance of deeper notions the player may be feeling. In a review for the news site *GamesRadar* [22] Heather Wald writes:

> "Night in the Woods has given me something to hold onto. My life can be a big, magnificent, scary mess. But in those moments when life loses its shape, and my anxiety and self-doubt resurfaces, Mae's nightmare eyes come into focus and I feel less alone. I feel reassured."

Figure 1.3: Screenshot of *Night in the Woods* portraying Mae and her friends [23]

Adventures with Anxiety!

Adventures with Anxiety! by Nicky Chase, released in 2019, is an interactive story about anxiety, in which the player acts as the anxiety of a woman [24]. Anxiety is personified as a red wolf that tries to protect the woman and will have something to say about every situation they are in and advise her against anything that could be remotely dangerous, physically or mentally. In this capacity, the wolf deprives the woman of most human interaction. He makes life for her noticeably harder by doubting everyone's intention and overestimating the dangers of every situation. The game is relevant to this thesis because of the way it thematizes anxiety. It depicts fear as a safety mechanism by the human body that, in this instance, got excessive. *Adventures with Anxiety!* acknowledges the way anxiety can hinder the affected person in their daily life and the hurdles it sets up, but aims not to remove the anxiety entirely but to experience it in moderation and when the situation calls for it [25].

Figure 1.4: Interaction with anxiety in "Adventures with Anxiety" [26]

1.1.3 Simluations

Video simulations are a particular type of media depicting Mental Illness. These short films are often available on YouTube and attempt to visualize symptoms of various mental Illnesses by simulating auditory and visual phenomena that take form when experiencing the illness. Typical Illnesses that are replicated are, e.g., schizophrenia, depression, or several anxiety disorders.

Due to the nature of the game accompanying this thesis, specifically the visual portrayal of anxiety symptoms, these simulations are relevant and can be viewed as an inspiration. An example of such a simulation can be seen in 1.5. The simulations work predominantly with distortions and changes in color values to visualize the symptoms. For this thesis, the approach is very similar; thus, the simulations will act as a source of inspiration. It is important to note that the reception of the accuracy of these simulations is disputed as they are most often made from a subjective perspective by an affected person and work with limited tools to create the effects [27]. The simulations are scarcely attached to any sort of exposition concerning the type of mental disorder and, regarding [6], are only one-half of a comprehensive discussion of mental disorder.

Figure 1.5: Screenshot of a Schizophrenia simulation found on YouTube [28]

2 Fundamentals

This chapter provides the fundamental terms and concepts necessary for understanding the development of the game prototype. It is divided into technical and psychological fundamentals.

2.1 Technical fundamentals

The game prototype accompanying this thesis was developed in Unreal Engine 5.3.2. This section explains the software and technical processes relevant to the programming and implementation of visual particularities of the prototype.

2.1.1 Unreal Engine 5

The Unreal Engine is a game engine developed and maintained by Epic Games, which was first used for their first-person shooter *Unreal* and later released for public use [29]. It has since been widely used in games and animation for television. Commercially successful examples are the before mentioned video game *Hellblade: Senua's Sacrifice* [30] and the *Star Wars* series *The Mandalorian* [31]. The Engine is free of charge for educational institutions and for developing projects that do not exceed a revenue of one million USD [32]. Unreal Engine supports the portability of desktop, mobile, console, and virtual reality platforms. While its coding language is C++, the logic of the program can also be realized in the engine's visual scripting system, *Blueprints*. The Blueprints Visual Scripting system is a node-based interface that allows for the creation of different gameplay elements, like mechanics and interactions. The nodes represent events that can be called functions and variables. Further, blueprints inherit nodes that control the flow of the different elements, like branches, loops, and sequences [33].

Materials

Materials define the appearance of actors in Unreal Engine. These assets can control parameters like color, texture, metallic properties, and roughness. To imitate surface details of an object, *Normal maps* can be applied to a material's property. Normal maps can be generated in different graphics programs and store information about how light is supposed

to interact with the surface. They are versatile tools that are used for the creation of surfaces, landscapes, post-processing effects, and visual effects alike. Inside the material editor, different parameters are set in order to finetune the different material properties [34]. In Unreal Engine, materials are created in its material editor based on the engine's node-based system. For this project, materials are essential to create visual effects called shaders, which are elaborated on in this section. In order to fully grasp their development process, some expressions used in the material editor need to be mentioned [35].

- *SceneTexture* expressions return a specified aspect of the scene specified by the user. Choosing *PostProcessInput0* will output the current pixel's color, whereas *GBuffer* related aspects will return the certain feature without lighting information to compute shadows. For instance, selecting *BaseColor (as stored in GBuffer)* will return the base color without lighting information [36].
- The *SceneDepth* expression is used by translucent materials to determine the existing scene depth at any location.
- In a similar fashion, the *PixelDepth* expression returns a pixel's depth or distance from the camera while it is being rendered.
- The expression *LinearInterpolate* blends between two inputs called *A and B*. It uses a third input called *Alpha* as a mask in order to transition between the two of them. The *Alpha* input declares how the distribution of the two inputs is weighted. If the value put for Alpha is 0.0, only the first input is used, and if it is 1.0, the second is used. Every value in between these numbers will achieve a blend of the two.
- The *Desaturation* expression desaturates an input to a degree set by the user or converts the colors into shades of gray.

Several other material expressions were used that execute standard mathematical operations like *Multiply*, *Subtract*, and *Add*. Further explanations concerning specific material expressions can be found in the official guideline by Epic Games at [35]. Additionally, by creating a *Material Instance* of a material, its values can be accessed and changed externally without compiling the material itself.

Post Processing

The term post-processing refers to a set of operations that allow for the enhancement of game graphics. In the Unreal Engine, post-process effects can alter the look and feel of a scene. A post-process volume actor must be added to apply the effects to a level. Adding this actor will allow for controlling visual effects that allow more finetuning of the scene and define how the player will perceive their surroundings [36]. The following effects that a post-processing volume can utilize find use in this project [37]:

- *Bloom* mirrors a lighting artifact of real-world cameras by distributing a glow around lights and reflective surfaces.
- A *Lens flare* effect describes the simulation of scattered light when viewing bright light sources.
- Enabling the *Vignette* effect fades out the image towards the edges.
- The *Motion Blur* effect blurs all moving objects. Objects moving faster will be blurred more severely.
- A *Film Grain* effect will distribute small particles across the screen and mimic old photographic films.

Niagara Systems

Niagara is a visual effects system added to the engine in version 5.0. Using Niagara, elaborate particle systems can be created using existing templates or by starting from scratch. The tool has different components that can be adjusted to achieve the desired look. These are, but are not limited to, the *Emitter Update*, which, for instance, defines the spawn rate of particles, and the *Particle Spawn*, which defines initialization details of the particles, like location, color, and size. Further, the *Particle Update* component decides properties like if the particle should change color over time or if the particles are affected by gravity or curl noise. *Niagara Systems* can be added to other actors and activated by blueprints [38]. In 4.1.3, the usage of *Niagara systems* to achieve a specific visual effect for the enemies is explained.

Third-Person Template

When creating a new project in Unreal Engine, the user can choose from a list of templates, one of which is the *Third-Person Template* [39]. The template was selected for this game prototype as it contains valuable assets that provide the basic functionality for the player‘s character, involving a mannequin model that the player can control. Basic movement animations, such as walking and jumping, have already been implemented. A camera follows the character to enable a third-person perspective. The mannequin model was slightly altered and used as the character for the protagonist, which will be explained later in 3.2.3.

User widgets

User widgets are blueprint classes that harbor functions for creating and updating user interfaces throughout a game. Further, the blueprint inherits an editor called *Unreal Motion Graphics UI Designer*, short *UMG*, that allows users to add different elements to the user

interface and set up their relation to each other. Common elements of these widgets are buttons, texts, bars, or boxes that group objects with each other. In a second tab, the *Blueprint Visual Scripting* handles all functionality of particular objects of the widget, such as buttons or text elements [40].

Level Sequences

Level Sequences are assets the engine provides to create cinematic sequences using the actors of a scene. In the different tracks of the editor, things like audio, effects, or actors can be modified in various ways. Actors are moved by setting keyframes at different time frames and changing the actor's position throughout. Also, the user can use the same method to adjust an audio track's volume, pitch, or other properties. *Level sequences* are primarily used to create *cutscenes*, which are small clips played in games to tell a story or to display short animations, like moving platforms [41].

Epic Marketplace

The Unreal Engine Marketplace is a platform hosted by Epic Games that provides digital resources for the development in the Unreal Engine [42]. These resources encompass textures, 3D objects, animations, sound effects, and entire projects. While Epic Games provides some of the assets, private users can upload and offer their products independently. The benefit of acquiring assets through the marketplace is that it is comparably easy to integrate most of them. Sellers of character models often specify whether their model is rigged to the standard Epic Skeleton, enabling compatibility with its animations without any modification. This benefit was exploited during the development of the prototype, as creating custom models or animations can be very time-consuming [42].

2.1.2 Shaders

Shaders are the processes used to implement Post-Processing effects in video games. They are hardware or software modules that enable the creation of rendering effects and manipulate how images are displayed on the screen. Shaders need to be deployed in real-time and consist of a large number of separate tasks that need to be executed at the same time. Thus, shaders are deployed on the GPU, which is short for the Graphics Processing Unit, and can compute these tasks efficiently due to its highly parallel architecture. The purpose of using shaders is to create visual effects, enhance the overall look of a scene, and improve performance [43]. For this thesis, shaders are used to mimic a particular art style discussed in 3.5 and 4.3.

2.1.3 Adobe Illustrator

Adobe Illustrator is an application used to create digital artwork, and its primary feature is the utilization of *vector graphics*. *Vector graphics* are a set of polygons that act according to mathematical formulas. The most notable feature is that a vector image can be scaled without losing quality. Adobe Illustrator is widely used to create illustrations like logos that require clean, sharp edges and scalability. For this project, *Adobe Illustrator* is used to create the user interface and its elements [44].

2.2 Psychological fundamentals

This section elaborates on the pathology, symptoms, and therapy of different anxiety disorders. Further, it will detail symptoms and other relevant terms used throughout the thesis and the accompanying game prototype.

2.2.1 Anxiety Disorder

An anxiety disorder is a type of mental illness that causes the subject to experience fear in excessive amounts and is most often accompanied by physical symptoms. While certain levels of anxiety are an everyday experience for anyone, people suffering from the disorder will sustain the accompanying symptoms for a prolonged period and at elevated levels [45]. The function of anxiety is to signal danger and conflict and to initialize a response. In the face of perceived danger, a person's body releases chemicals, the heart rate and blood pressure rise, the bronchial tubes expand, and the metabolism speeds up [46].
Anxiety disorders do not have one singular etiology and can either be of biological, psychological, or social cause [47]. There is a higher chance of an anxiety disorder for people directly related to a person with an anxiety disorder than in the relatives of someone without an anxiety disorder.
While they can be treated, anxiety disorders are kept up by a cycle, seen in figure 2.1 that starts by feeling anxious and results in being overly aware of dangers or physical symptoms. The person will avoid things, places, and situations that could cause the dangers mentioned earlier. This will result in short-term relief that will eventually set the long-term reassurance that avoiding these situations keeps them safe. In return, the perceived anxiety caused by these triggers will gain strength, which will make it increasingly more complex to break out of the cycle [48].

Further, anxiety can develop due to internal and external stress exceeding the subject‘s means of handling stressful situations. Social factors that work towards developing an anxiety disorder can be experiences like the death of a significant person, divorce, life-altering circumstances like job loss, or illness. Exposition to abuse, violence, terrorism, and poverty can also contribute to a particular susceptibility to anxiety disorders [49].

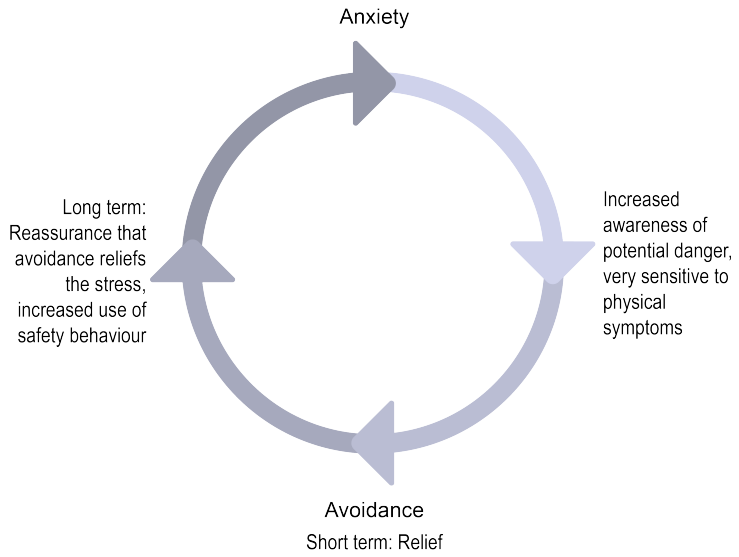

Figure 2.1: Loop of anxiety, inspired by [48]

Anxiety disorder can be categorized into different types with different clinical symptoms. Most commonly known are:

- Generalized Anxiety Disorder
- Phobias
- Agoraphobia
- Panic Disorder
- Post-Traumatic Stress Disorder
- Obsessive Compulsive Disorder

A phobia is a type of anxiety disorder that causes an uncontrollable, lasting fear of a particular object, situation, place, or activity. Most often, people suffering from the illness will avoid the source or places connected to this fear. Phobias are connected to panic attacks, which can occur when the afflicted person is confronted with the source. Examples of phobias include but are not limited to, flying, dogs, tunnels, and heights. A specific variant of phobia is agoraphobia, which involves the fear of being in confined places that are hard to escape. Further, they will often avoid crowds. People with an obsessive-compulsive disorder show patterns of repetitive thoughts and behaviors that interfere significantly with normal occupational functioning, social activities, or relationships [49]. A Panic disorder makes a person feel elevated levels of anxiety, stress, and panic overall. The most defining symptom of the illness is that someone who has a panic disorder will have frequent and seemingly arbitrary

panic attacks, even when there may not be an apparent trigger for it. Panic attacks often include physical symptoms that the subject may mistake for severe physical conditions [50]. Although they might be similar in terms of symptoms, it is essential to differentiate between a panic disorder and an anxiety disorder, as it is often seen as synonymous with each other, even though panic disorder is a specific type of anxiety disorder. While people with any anxiety disorder might experience panic attacks, even frequently, there will be triggers that start them. People who have a panic disorder will have them on seemingly no grounds. In [51], a differentiation is made as follows:

> "Panic disorder is an anxiety disorder that involves multiple unexpected panic attacks. A main feature of panic disorder is that the attacks usually happen without warning and are not due to another mental health or physical condition. There is often not a specific trigger for them. Not everyone who experiences a panic attack develops panic disorder."

This thesis and the project it addresses center around generalized anxiety disorder, the most common type of anxiety disorder [52]. Thus, it will only thematize the associated symptoms, behavior, and therapy for this particular variant of the illness. In the following, the term anxiety disorder will be used for this variant exclusively. Generalized anxiety is characterized by excessive and uncontrollable worrying. The disorder is often already developing in childhood or the teen years and can be inherited [53]. Generalized anxiety disorder is considered a chronic illness [54].

Symptoms

Symptoms of a generalized anxiety disorder are individual to the subject and may appear in different forms. It should also be noted that the severity and frequency of these symptoms can vary. In the ICD-10 [55], an official classification of mental disorders by the World Health Organization, the following criteria must be met for a diagnosis:

- There must have been a period of at least 6 months with prominent tension, worry, and feelings of apprehension about everyday events and problems.
- At least four of the symptoms listed below must be present, at least one of which must be from items 1 to 4:
  - (1) palpitations or pounding heart, or accelerated heart rate;
  - (2) sweating;
  - (3) trembling or shaking;
  - (4) dry mouth (not due to medication or dehydration);
  - (5) difficulty in breathing;
  - (6) feeling of choking;

(7) chest pain or discomfort;

(8) nausea or abdominal distress (e.g. churning in stomach);

(9) feeling dizzy, unsteady, faint, or light-headed;

(10) feelings that objects are unreal (derealization), or that the self is distant or "not really here" (depersonalization);

(11) fear of losing control, "going crazy", or passing out;

(12) fear of dying;

(13) hot flushes or cold chills;

(14) numbness or tingling sensations;

(15) muscle tensions or aches and pains;

(16) restlessness and inability to relax;

(17) feeling keyed up, on edge, or mentally tense;

(18) a sensation of a lump in the throat, or difficulty swallowing;

(19) exaggerated response to minor surprises or being startled;

(20) difficulty in concentrating, or mind "going blank", because of worrying or anxiety;

(21) persistent irritability;

(22) difficulty in getting to sleep because of worrying;

- The disorder does not meet the criteria for panic disorder, phobic disorder, anxiety disorders, obsessive-compulsive disorder, or hypochondriacal disorder.
- The anxiety disorder is not due to a physical disorder, such as hyperthyroidism, an organic mental disorder, or a psychoactive substance-related disorder, such as excess consumption of amphetamine-like substances or withdrawal from benzodiazepines.

Not all symptoms will be implemented into the game in the upcoming prototype. That means that the depiction might be vastly different from the experience of some people with an anxiety disorder. Symptoms found in the game are sensory overload, derealization, and depersonalization, which will be explained more in-depth in the following section.
Derealization is a symptom of generalized anxiety disorder. It conveys a feeling of detachment from the environment and people around oneself. It is often described as *dream-like* or as the world turning *seemingly fake*. This state of seeing the world as distorted can last from a few minutes to months.
While people experiencing derealization describe the feeling as scary and unnatural, the symptom is not associated with any type of psychosis [56].

Depersonalization describes the feeling of being detached from one's own body. The people affected experience a distorted self-image that can manifest by feeling like parts of one's body are enlarged or shrunken, or overall distorted. Further, it is often described as feeling *robot-like* or feeling like one's memories are not their own [57]. Both abovementioned symptoms can stem from a variety of factors, like childhood trauma or usage of drugs. A common trigger for these symptoms is an ongoing anxiety disorder, which utilizes these symptoms as a way to shield the person with anxiety. [57].
Sensory overload is a state in which the person experiencing it will become overwhelmed by more stimuli in their five senses than they can process. They will experience increased levels of irritability, stress, and physical discomfort. During this state, it will be harder to slow down and unwind. The urge to block out all input, like volume or touch, will get increasingly more present. The causes for this state can include but are not limited to, crowds, loud noises, sudden or unwanted physical contact, or sudden light changes. People with a generalized anxiety disorder will have a higher susceptibility to sensory overload [58].

A further state of mind that can occur to a person with an anxiety disorder is a general feeling of emptiness. The state can be caused by experiencing trauma or different mental illnesses, including anxiety disorders. Among other things, it is maintained by suppressing emotions. Thus, one crucial aspect of trying to overcome this state is to experience one's feelings and observe them mindfully [59].

Therapy

In order to treat an anxiety disorder, the person suffering from it will have to partake in psychotherapy. The most successful form of therapy is cognitive-behavioral therapy [60]. The goal is to change how the patient approaches anxiety-inducing situations. The therapy can be divided into two parts. During the cognitive part, the thoughts of the patient are observed. By critically contemplating learned thought patterns that trigger anxiety, unrealistic thoughts and fears can be identified. By repeating this process, the patient will eventually learn to recognize these subjective thoughts by themselves and thus control them better. In the second part, the fear is being confronted. Alongside learning techniques to cope with anxiety, the patient will slowly be introduced to fear-inducing scenarios. Initially, it may only be through images, audio, or dialogue or by letting go of control to a certain amount. This process will gradually increase until the patient is equipped to cope with the fear on their own[49] [45].

3 Conception

As a foundation for this thesis, a prototype of an action-adventure game was developed that intertwines themes of generalized anxiety disorder and gameplay. The action-adventure genre is known to insert the player into situational problems and add a strong focus on exploring the setting. It generally mixes concepts of action games, like physical skills and combat, and adventure games, like puzzles. Famous examples of the genre are *The Legend of Zelda* or *God of War* [61]. This chapter lists conceptual choices made before the game's development and clarifies the reasoning behind these choices.

3.1 Idea

To illustrate the concepts mentioned above, a game with the title *Eye of the Storm* was created. Its goal is not to romanticize or catastrophize mental illness in any form but to create a nuanced perspective of the topic in an interactive medium. As mentioned in 1.1.2, these perspectives are lacking in most video games. The game aims to present mental illness not necessarily as positive but as something natural that can pass while also depicting the protective function of specific symptoms. It is supposed to debate the topic positively as a guide for people who have not been introduced to it and can see themselves in the issue. In the game, the main character struggles with a generalized anxiety disorder and is coming to terms with it throughout the game. The game falls under the 3D action-adventure genre and uses the third-person perspective.

3.2 Story

A simple story that leads the player through the levels gives the game a leitmotif. The game starts with the main character at a party where a bass track, a talking crowd, and a simple electronic music track can be heard. After a while, the main character's heartbeat can be heard and will get increasingly louder over time. Accompanying, a ringing and static noise become increasingly more apparent. The soundscape intensifies until everything stops at once, and the screen goes black. Afterward, the main character finds themselves in a nearly empty white space without any details, from now on called *the void*. An individual greets them and provides an exposition of their situation. Its function is to act as a guide to the player. The guide proclaims that they must confront their emotions to overcome this state.

The main character will then head into the first level, representing an emotion: fear. They have to defeat enemies using a sword and abilities that represent symptoms of an anxiety disorder. After completing the level, the main character returns to the void where some of the fear stage's environment has overtaken the void, representing the acceptance of the emotion. The same will happen when the main character enters the second stage, which represents the emotion of anger. After returning from the anger stage, the guide encourages the player to look around and see their progress. The protagonist takes a moment to observe the emotions that have already overtaken the void and is told to take a break, ending the prototype of *Eye of the Storm*.

3.2.1 Dialogue

At various points throughout the game, the guide and the protagonist converse with each other. During these dialogues, seen in 3.1, assumptions about their respective personalities can be made by paying attention to their way of speaking and the content of their responses. This is elaborated further in the respective introductions of the protagonist and the guide, which can be found in 3.2.3. In some instances, the player can choose from an array of responses, and the guide's answer will vary accordingly. There are fixed points in the story progression at which the guide will approach the player automatically, which are held to further the story. If the player interacts with the guide apart from these encounters, the guide will respond with simple one-lined statements.

Figure 3.1: An interaction between the protagonist and the guide

3.2.2 Levels

The structure of the game's world is modeled after the *Hub-and-spoke model*, which often can be found in video games and can be seen in figure 3.2. This approach of structuring levels declares one hub as the center of the game, where the player will return after each level or otherwise defined game section [62] [63]. The void functions as the hub, as the player returns after each emotion stage. All assets connected to the environment of the levels are part of several asset packs provided at the *Unreal Engine Marketplace* [64] [65] [66] [67].

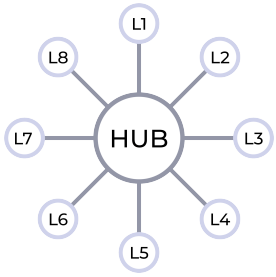

Figure 3.2: Hub-and-spoke model inspired by [63]

The initial vision of the game was to add six levels, each representing an emotion. A model of the *emotion wheel*, seen in figure 3.3, would have been utilized as a base. Emotion wheels are often used in therapy to help patients express their emotions by name [68].

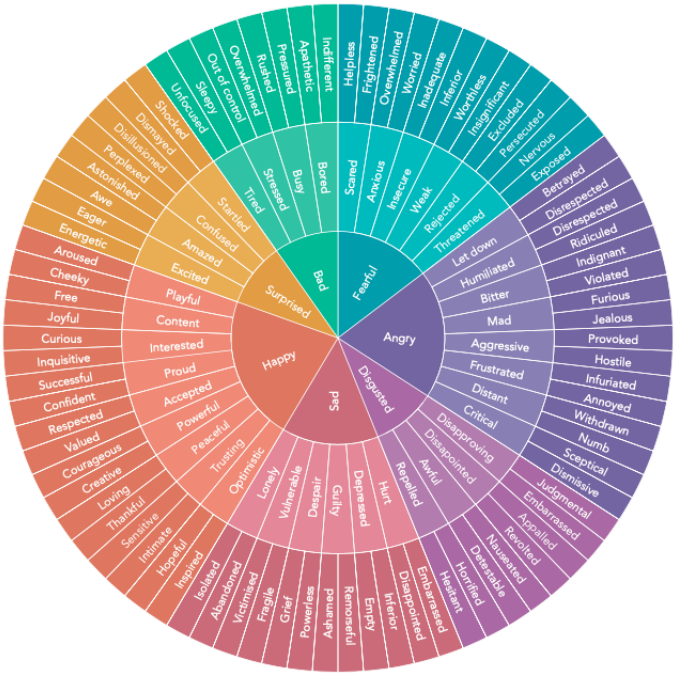

Figure 3.3: An exemplary emotion wheel used in therapy [68]

These levels would have been structured like the two current levels. For instance, a stage representing the emotion of sadness would inherit muted colors and a bleak environment. Due to the intended size of this thesis, anger and fear were chosen as they can be depicted through very distinct color schemes and tonalities. The two levels, as well as additional scenes that are intended to round out the game, are introduced in this section.

The void

The void can be seen as a node between the different stages and a blank canvas that slowly begins to gain character throughout the game. As can be gathered in figure 3.4, the space of the level starts completely blank but gains some of the environment of each level the player successfully overcomes. It represents a feeling of emptiness that people with anxiety often report and is discussed further in 2.2.1. After experiencing a panic attack at the party, where there is flashing light and loud music, the void's contrast is intense and takes the protagonist by surprise. In the beginning, the level is empty other than the portals that take the player to the different stages and the guide. While in the void, soft music is playing that is elaborated upon in 3.7. Based on the void acting as a hub to other, more active levels, the name *Eye of the Storm* was chosen for the game. The feeling of emptiness does not necessarily mean there are no emotions worth exploring; instead, there are quite a few that the person shields themselves from. Hence, the player is standing in the eye of the metaphorical storm while the emotion levels represent the storm around them.

Figure 3.4: The void

Fear

The first level, *fear*, strives to feel cold and capture the spirit of fear itself. The landscape, as seen in figure 3.5, is rocky, has steep cliffs, and displays a forest setting, which is meant to invoke an uneasy feeling. The borders of the level are blocked by large, gray boulders overrun by crystals. The high borders of the level, as well as its cliffs and valley, aim to make the player feel constricted. In general, the environment is designed to be unsettling and trigger fear in the player. Throughout the level, cold tones of blue and gray are applied to the floor and its scene actors. All over the stage, blue foliage in the form of flowers, mushrooms, and grass decorates the floor. Multiple enemy groups are placed throughout the stage. These enemies are equipped with colors that match the blue tint of the environment. No human structures can be found in the fear stage except for one large gate at the end of the level that leads them to the portal back to the void. This is due to the intent of creating a fear-driven atmosphere, as human structures might be associated with familiarity and safety. At some points of the stage, the scene is covered in fog to convey a feeling of being lost. At the end of the stage, blue fireflies fly around to greet the player who has now reached safety.

Figure 3.5: The fear stage

Anger

The *anger stage*, seen in figure 3.6, is modeled after a desert; its landscape is a large, desolate plane. In contrast to the fear stage, there are no hills or pathways. This illustrates the emotion of anger taking up much space in a person‘s mind. Human structures like tiles, pillars, and ruins of former halls can be found throughout the level. While contrary to the fear stage, human structures are all over the level; they are found in a ruined state. The

destroyed state represents the destructiveness of rage and anger. Foliage in the anger stage includes cacti, grass, flowers in a warm color scheme, and stones. The borders of the level are not concealed by any large structures, like they are in the fear stage, but are blocked by invisible walls with a large amount of landscape still to come. By doing so, it looks like the desert goes on indefinitely and aims to comment on how easily one can get lost in one's anger. To prevent the player from running into the wall and potentially compromising the game's immersion, a pop-up nudges them not to lose themselves in their anger and turn back. At one point in the production, the idea was to add a heat flicker effect to the screen to underline the heat theme. However, it was swiftly abandoned since the effect would lessen the impact of the symptom abilities that will be introduced in 3.4 would have. Like in the fear stage, at the end of the level, the player is embraced by fireflies that are in the area around the portal back to the void. In the anger stage, the colors are more intense and piercing, with red being a central component in the overall color scheme. The several groups of enemies comprise the same enemies introduced in the fear stage, with the difference being tinted red this time.

Figure 3.6: The anger stage

Additional levels

Additionally to the emotion stages and the void, two further levels each fulfill a specific purpose. First, a short party sequence plays at the start of the game. The protagonist is not equipped to handle the many stimuli of the party and experiences a panic attack, which is portrayed by relying on auditory cues in the form of tinnitus and a beating heart. The sounds get increasingly louder and eventually overshadow the sounds of the party until the whole soundscape falls silent. No actual party is represented; it is depicted merely through

flashing lights that change color rapidly and loud party sounds. It is not specified what type of party it is or why the player is participating, as the scene is solely added to set the stage and give an identifiable reason for the protagonist to have the strong reaction that leads him to the void.
Further, a showcase can be accessed through the main menu. Using three buttons on the simple user interface, the player can rotate through the three symptoms of a generalized anxiety disorder and see their visual representation on screen without the gameplay element. The symptoms, in combination with their respective abilities, are discussed in 3.4. As the player hovers over the three buttons, a small window, also known as *tooltip*, will appear on the screen and provide details about the specific symptom. The tooltips are added to increase the exposition the player receives concerning generalized anxiety disorder and to ensure they understand the reason for the symptom‘s existence. This notion is based on the study in [6] and its findings that stigmata are best dissolved by providing information and any type of symptom visualization.

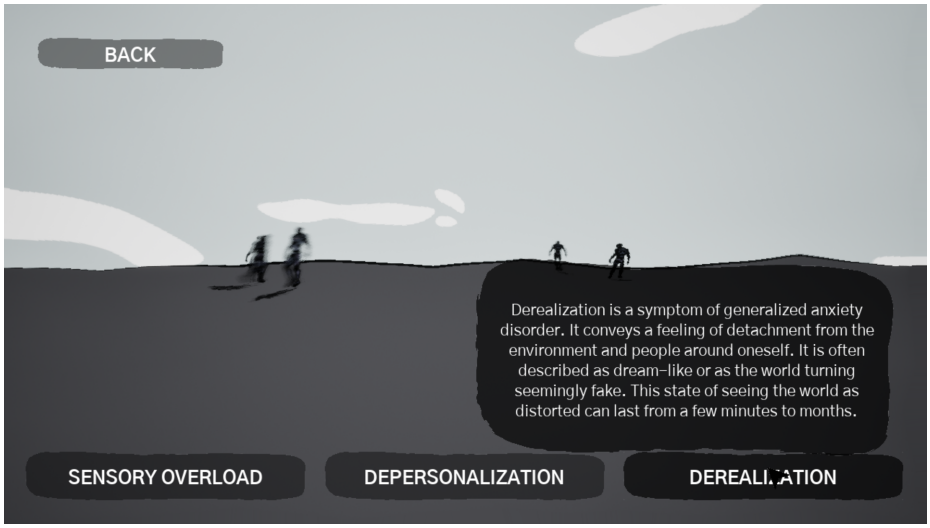

Figure 3.7: The showcase level displaying the visual representation of certain symptoms

3.2.3 Characters

Two characters are featured in the game and interact at various points throughout the story. This section introduces and briefly characterizations these characters.

The protagonist

The player and protagonist in the story is an unnamed person experiencing a panic attack during a party. Triggered by this, they come to their senses in the void. There, they will meet a guide that helps them understand their emotions and embark on a quest to experience and understand them thoroughly. Assumptions about the protagonist‘s personality and demeanor can be made by analyzing the dialogue between them and the guide. A full extract of the dialogue can be found in the file *GameplayElements_ Dialogue.pdf* on the USB drive attached to the thesis. They are portrayed as brave and open to confront their emotions. When interacting with the guide, they quickly accept their situation and can even make humorous remarks about it. While they appear to accept the circumstances, some dialogue options the player can choose from are not as cheerful, and other choices display clear signs of exhaustion. The protagonist is displayed as a complex character trapped in a severe case of anxiety.
The character‘s appearance is very simple on purpose with the intention that the player can insert their own personality into them properly. The standard Unreal Engine model, equipped with a basic gray tint, created the best surface for the player‘s projection. This is especially important since, to act in line with the eudaimonic approach to media, the player should look inward and, if possible, apply the game‘s subjects to themselves.

The Guide

The guide, seen in 3.8, is a character whose role is to help the main character navigate their emotions by giving directions and affirming words. Their first appearance is at the start of the game when the main character arrives in the void. They brief them on the circumstances and explain their next steps, which are moving on to the fear stage and thus confronting their emotions. The guide will appear on several occasions to encourage the player or gently urge them to pursue their upcoming challenges.
By reading reading from the dialogue in the file *GameplayElements_ Dialogue.pdf*, their personality can also be approximated. Their way of speaking is vivacious and encouraging, and they interact with the main character playfully to ensure that the topic never becomes too much of a burden while also avoiding downplaying the situation. Further, they act as a metaphor for *structural dissociation*, which describes the process of dividing oneself‘s internal conflicts between the *normal self* and the *traumatized self*. The conflict can lead to identity confusion and a deep sense of emptiness and is often a result of narcissistic abuse [69]. In this highly abstract visualization of the coping mechanisms, the guide acts as the normal self that helps the traumatized self through a painful situation. The skeletal mesh of the guide was acquired from the *Epic Games Marketplace* [70].

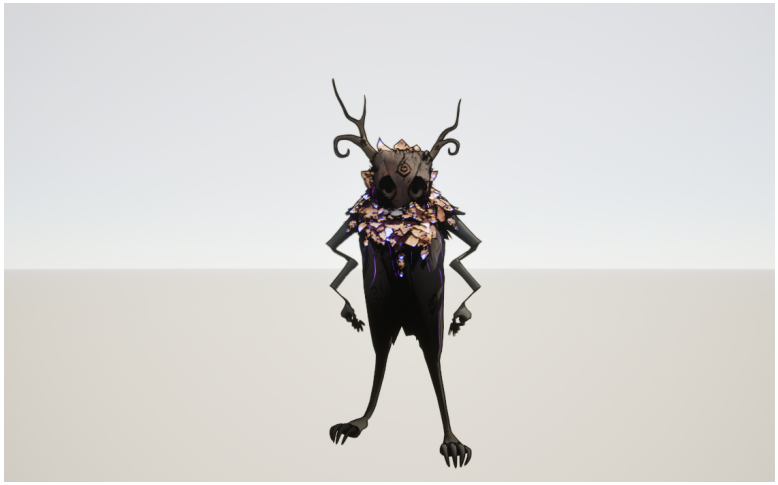

Figure 3.8: The guide

Enemies

Enemies come in two forms depending on the stage they are in. They appear to the player as stone golems, humanoid creatures made of stones or similar material with colossal strength. The enemies match the stage's overall color composition. All enemies inherit a blue tint throughout the fear stage, while they appear in a bright red tint in the anger stage. Instead of a sword, the golems use their hands, which are made of stone, to attack the player. While there is plenty of mythology around the golem creature, it was primarily chosen because of its neutral appearance, which matched the game's overall look rather than being a metaphor for anything more profound. This also allowed for slight alterations of the appearance in terms of color and, thus, added the intended individuality to each stage. The skeletal meshes of the golems were acquired from the *Epic Games Marketplace* [71]. The golems can be seen in 3.9.

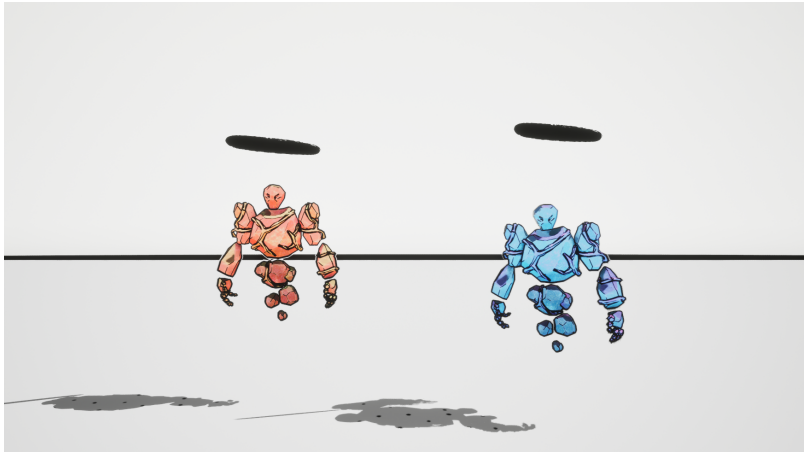

Figure 3.9: Both enemy types

3.3 Mechanics

The player has several mechanics to move through their environment that fall into different categories. First, there are locomotion mechanics. As is standard in this game genre, these are running, sprinting, jumping, and crouching. Additionally, they can vault over small objects that are approximately waist-high. Lastly, they can dodge by rolling in the direction they are looking. Running, vaulting, and dodging use up stamina, displayed to the player via a user interface introduced in 3.6. Stamina will continuously recharge while the player is not performing either of the three actions.
In combat, the player can lock onto enemies to increase their chance of hitting the target. They can attack an enemy using a sword they receive at the beginning of each stage. The sword acquired in the second level carries out slightly more damage. If there had been more stages, this would have continued throughout the game, with each sword dealing more damage than the previous. To assist the player in combat, they have a selection of abilities that will be explained further in 3.4.

3.4 Depiction of Anxiety

Studies have shown the positive effects of incorporating interactive media into the attempt to destigmatize Mental Illnesses. While it can considerably improve knowledge about these illnesses, there are concerns about using these media types without offering any education about the topic at hand. This criticism is essential so that there will not be a significant focus on the symptoms without trying to connect with the person experiencing them [6]. To prevent this effect, the game offers exposition to the topics thematized, which will be mentioned in this section. The game's central theme is the depiction of a Generalized Anxiety Disorder. Thus, its presence can be seen in various aspects of the game, which are discussed in the following section.

Symptom abilities

During combat, the player can use *Symptom abilities*, which enhance certain features of their movement, vision, health, and stamina. They are accompanied by audiovisual feedback that mirrors the symptoms mentioned in 2.2.1. An overview of the three symptoms, with their respective auditory, visual, and gameplay representation, can be found in 3.1. It is important to note that due to the engine's limitations, a lack of user studies, and time constraints, not every aspect of the symptoms can be fully visualized, and the accuracy cannot be verified. Only one ability at a time can be performed, and after its activation, the effect is in place for ten seconds before the game returns to its normal state. Afterward, the player must wait for five seconds before they can use one of the abilities again. The type of ability that is currently equipped can be changed by scrolling the mouse wheel up or down to ensure a

quick switch during potentially stressful combat situations. Initially, a radial menu was used to switch between the abilities. However, it was swiftly changed to the current mapping due to an evident lack of intuitiveness and difficulty operating.
The following traits of a generalized anxiety disorder are implemented as symptom abilities:

- Derealization
- Depersonalization
- Sensory Overload

These three symptoms, in particular, were chosen because their visual and auditory sensations are well-documented and consist of factors that can be visualized well without being too abstract. Other symptoms, like those that are more physical, are not as suitable due to the video game medium. In 3.1, the individual descriptions of the three symptoms and the intended realization in the game are listed.

Symptom	**Description**	**Visual**	**Auditory**	**Gameplay**
Derealization	Environment/People seem unreal, Feeling like in a dream, foggy	Enemies get blurred, Desaturated colors, Distortion, Motion Blur	Dampen sound, Lower volume	Slow Motion
Depersonalization	Feeling out of body, Loss of control, Feeling like a robot	Desaturated colors, Radial Blur, Third Person	-	Increased Field of View, Invincibility
Sensory overload	Too many stimuli, the situation gets increasingly stressful	Temperature increase, Vignette, Static, Bloom, Radial Blur, Motion Blur, Increased white intensity	Static, Tinnitus	No stamina exhaustion, Increased movement speed

Table 3.1: Anxiety symptoms and their in-game representation based on 2.2.1

The ability *Derealization* enables a slow motion on the enemy's part, while the player can continue to move at average speed. The ability aims to allow the player to reposition themselves or outplay the enemies in combat. If well-timed, the player can dodge incoming enemy attacks or quickly get in as many hits as possible before the time runs out. A desaturation of the scene visualizes the perception of people affected, in which their environment appears to be lifeless and pale. The implemented time dilation strives to represent the feeling of not being in synchronization with one's environment and the people around oneself. A distortion that moves whenever the player is also moving the camera mirrors the common conception of the environment being not actual and simulated and everything feeling like a dream. All enemies in current combat are equipped with a particle effect that hints at them fading away and blurs them to a certain extent. During the ability, the music's volume, as well as its pitch, is halved. This mirrors the sensation of noises being perceived as *out of place*.

Figure 3.10: The derealization ability

Depersonalization will leave the person experiencing it feeling like their actions are not their own, like they are removed from their body, and they are watching themselves from afar, often described as from above. Hence, the *Depersonalization ability* increases the distance between the player character and the third-person camera, enabling the player to perceive more of his situation to mirror the sensation as mentioned earlier. Additionally, they cannot be harmed by enemies for the duration of the ability to convey the feeling of not being in their own body. The ability allows for more straightforward combat and can be especially valuable in combat with more than one enemy when the possibility of evading all incoming damage becomes more unlikely. Additionally, perceiving more of the scene will aid in navigating through enemy crowds.

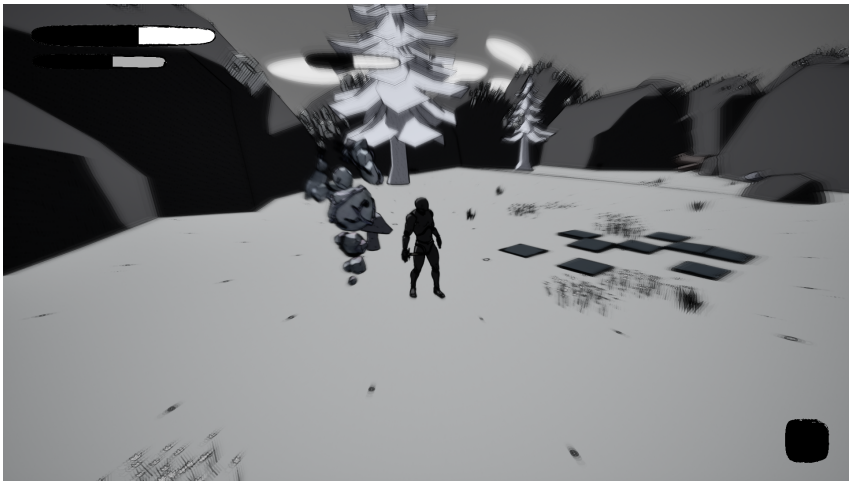

Figure 3.11: The depersonalization ability

When people talk about a *sensory overload*, they refer to the state that is most often triggered by anxiety or stressful situations where they are not able to process all incoming stimuli properly. As an ability, *Sensory overload* locks the player's stamina for its duration while boosting their movement speed significantly. The accompanying visual effects are elevated temperature levels, a faint grain, and a black vignette from the edges of the screen. Additionally, local exposure is modified so bright surfaces seem blinding. Regarding auditory feedback, tinnitus, and ear ringing start playing. While the ability is active, a motion blur is deployed to make the environment more challenging to grasp, an effect that is also often associated with experiencing a sensory overload. In general, the ability strives to mimic the feeling of not being able to properly process the stimuli of a scene and convey rising tensions within a person. The newfound ability to move quicker for the symptom ability's duration, as well as not consuming any stamina, alludes to the real-life counterpart of *going into fight-flight mode* and choosing to flee the stress-inducing situation.

Figure 3.12: The sensory overload ability

Using the abilities increases the stress level mentioned in 3.3. When the stress level peaks, the character will experience a shutdown and temporarily be out of action. This mechanic aims to prevent sending an inaccurate message or romanticizing anxiety. While acting as protection, these symptoms can still harm the one experiencing them, as mentioned in 2.2.1. The usage of anxiety symptoms as a way of progressing in the game is inspired by *Hellblade: Senua's Sacrifice* as it offers an approach to mental illness that not only renounces any catastrophizing undertones but depicts them as helpful. These symptoms might not be helpful long-term or aid in overcoming an anxiety disorder but offer temporary defense mechanisms created by one's mind and body [72].

3.5 Art Style

The primary inspiration for the stylized appearance of the game is the French artist and cartoonist Jean Giraud, also known under his nome de plume *Moebius*. Jean Giraud's influence can be found in many instances when looking at science fiction design. For instance, his comic *The Long Tomorrow* directly inspired the art direction of the movie Blade Runner. His highly surreal imagery most often displays imaginative worlds that explore philosophical questions. Jean Giraud's art style is defined by his distinct and clean linework and the flat coloring of objects throughout his illustrations. [73] Moebius' style of drawing falls under the term *Ligne Claire*, which was pioneered by the creator of *The Adventures of Tintin*, *Hergé*. The style is known for its clear outlines with close to no hatching and its lack of contrast and shading in colors. The characters and objects depicted are simplified in most cases [74]. Today, the art style he helped form can still be found in several media types. An example that was also used as an art reference for the game is the 2021 game *Sable* by the studio *Shedworks* [73]. This particular art style was chosen because of its heavily stylized nature, which allows for a clear way to detach the game's world from the real-life world and accurately display the dreamlike state the protagonist finds themselves in. In order to approximate the look of Moebius' comics, shaders were added to the game's scenes that are explained in 4.3. To replicate the Line Claire style, the objects throughout the levels only have one singular color value, with an exception for the game's characters. They are simplified, and any details found on them are scarce.

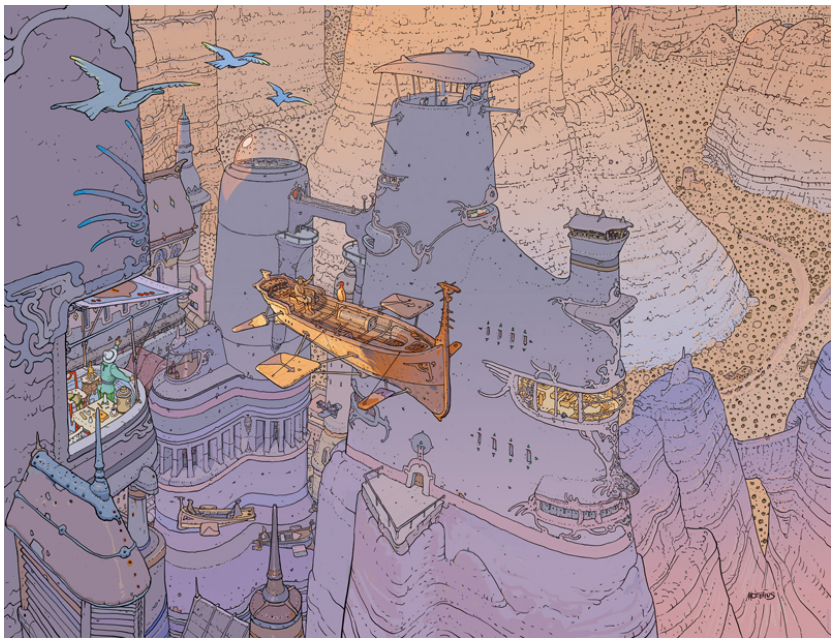

Figure 3.13: An illustration by Jean Giraud, aka Moebius [75]

As mentioned in 3.2.2, the color schemes of both levels represent the emotion it mirrors. The primary colors of the fear stage, seen in figure 3.14, are cold and mostly have a blue tint. While blue is associated with several things depending on culture and religion, blue was chosen because it can appear distant due to its association with coldness [76]. It is important to note that the association of colors with certain feelings, moods, or concepts is not based on any scientific foundation and is barely applied to create an overall understanding of the setting.

Figure 3.14: Color palette in the fear stage

The color red was chosen to represent the emotion of anger properly. Red is often associated with strength, passion, and confidence, but it also symbolizes anger and can be aggressive. Red is often used as a call to action [77]. Resultingly, the color is spread throughout the whole level and strives to invoke a feeling of rising temper and a constant state of being enraged. An extract of the level's color scheme can be seen in figure 3.15.

Figure 3.15: Color palette in the anger stage

3.6 User Interface

The user interface strives to relay all the information to the player that they need to experience the game properly. There are several instances where the player might need guidance concerning the upcoming points of action or how to use a particular object. In that regard, different introductory user interfaces were integrated alongside interactive interfaces that further the game. The graphic elements used throughout all user interfaces follow the same color, outline, and shape schemes to ensure high coherence. Since the color schemes of the stages vary enormously from one another, the colors of the user interface are black and white only, with some variations in terms of opacity. The different elements can be seen in figure 3.16. Further, more specific graphic elements of the user interface carry on the same color and outline schemes.

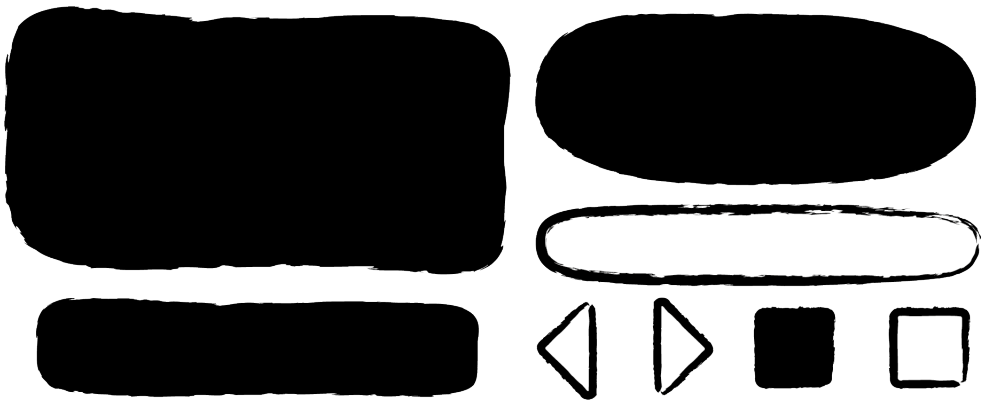

Figure 3.16: User Interface elements used in the game

Figure 3.17: Icons visualizing the three different symptom abilities

The user interface fulfills various tasks. First, a so-called *head-up-display*, as seen in figure 3.18, displays every information the player needs at any given time during the game; the head-up-display from here on is referred to as the *HUD*. A progress bar on the upper left side of the screen displays the player's health, and a progress bar under it references the player's stress level. Further, the currently equipped symptom ability can be found at the lower bottom right corner, visually representing its cooldown time. A stamina wheel is displayed with a slight offset to the left of the screen center. Opposed to the other HUD elements mentioned, the stamina wheel will disappear once it reaches its maximum value. It will only reappear once the player performs actions requiring stamina: sprinting, vaulting, or dodging, for instance. This stems from a standard guideline for designing video game user interfaces, which states that the player should only see the necessary information at the current point of the game [78]. The player will always need information about their health status to plan their next steps and evaluate if they should take specific actions, like pursuing an enemy while their health value is low. They also need information about their currently equipped symptom ability, so they will not use the wrong ability on accident and waste a whole cooldown cycle. The cooldown representation, however, can be detached from the HUD as soon as the bar is filled up, as the player will automatically know the ability can be used in the absence of the bar.

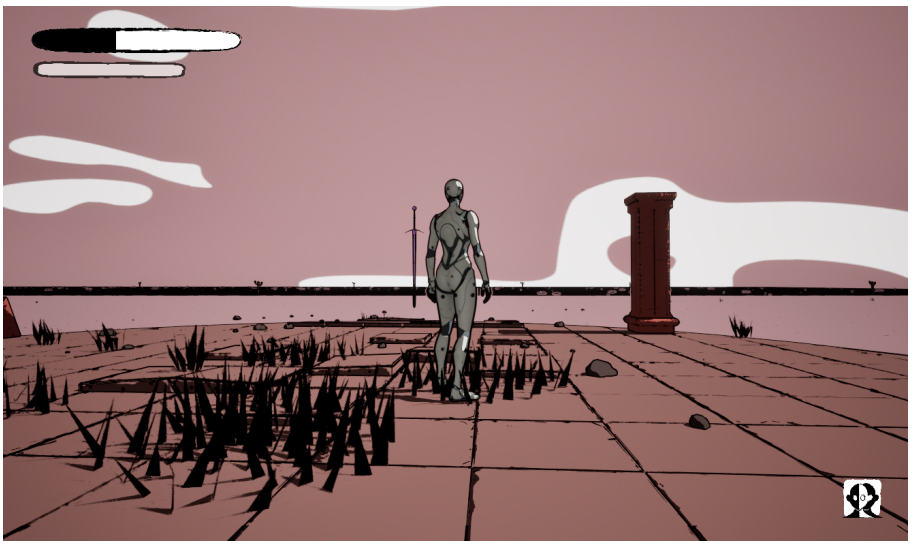

Figure 3.18: The *head-up-display* in-game

The game's dialogue is also shown inside a text container at the bottom of the screen. Whenever a character starts a conversation with the player, their dialogue will be displayed in the container. The dialogue can be driven forward by clicking on one of several buttons whose descriptions mirror the responses the player can return. In some instances, the player can choose from several responses; in others, they only have one. The dialogue interface will disappear once the conversation is over.
Further, throughout the game, the player will receive pop-ups that alert them to what is happening at that particular moment or to what special actions they can take. Such an example is the alert that is called upon when the player gets close to a sword that informs them to press the E key to pick it up, as seen in figure 3.19. Afterward, it displays a text that says *Press I to open up the inventory.* The player will receive additional pop-ups during the game, triggered when they get close to the map's border and when a stage is cleared.

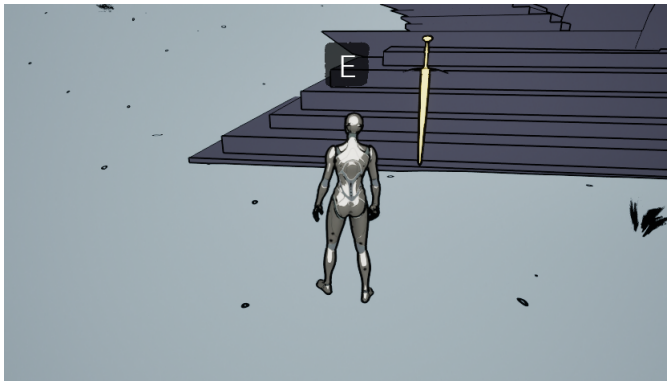

Figure 3.19: An user interface that instructs the player to press the E key

The game's main menu consists of the same graphic elements, seen in 3.16 and the rest of the game. Four buttons enable the player to start the game, open the credits screen and the settings, and quit the game. The background is made up of a solid color in high contrast to the black buttons. To reiterate the study in [6] that emphasizes providing information and context to create an understanding of various mental illnesses, an initial warning screen opens up right at the start of the game. The warning screen consists of a general trigger warning for topics like anxiety and trauma, which some players may find distressing. Further, a link leads to *findahelpline.com*, an international website that provides various mental health resources and educated contact persons.
A general tutorial interface will appear when the player heads toward the first stage. They can then navigate through it to gain insights about the controls and gameplay elements. This tutorial interface can be called upon at any given time by a specified key or by accessing it through the pause menu. Lastly, the player can access their inventory by pressing the *I* key. Weapons picked up at the current level can be equipped in a respective weapon slot. Along with the weapon slot, the inventory shows a live display of the main character. The inventory was implemented to add more than one weapon type to the game, a concept that was ultimately dismissed. However, due to its potential to add value to the game, the inventory remains even though the player won't need to switch weapons.

3.7 Sounds

Several sounds are used to complement certain aspects of the game. This section lists all scores and sound effects implemented.

3.7.1 Soundtrack

The ambient music consists of one composition explicitly made for this game by a fellow student, Michael Weiler. When commissioning the piece, the goal was to create music that captures the overall tonality of the project. It was supposed to sound profound, dreamlike, and melancholic but also hopeful and calming to the same extent. Ultimately, it was supposed to walk the line between the two. The soundtracks to other games like *Gris* and *The Legend of Zelda: Breath of Wild* were used as inspiration.
This composition slowly builds itself from level to level. Initially, only a vocal and a bass track can be heard after first arriving in the void. Three new tracks are played during the fear stage: a synth, chords, and hammered dulcimer. The second time the player arrives in the void, these three tracks are added to the initial two tracks. The overall score sounds slightly more complete. Repeating this, three new tracks are introduced in the anger stage. After completion, these remaining three tracks, consisting of guitar, arp, and brass, complete the score. The purpose of layering these tracks and introducing them in waves is to compliment

the environment that each stage adds to the void and to emphasize the intended feeling of accomplishment when one of the emotions returns to the main character's subconsciousness. The score's goal is to set a calm and mellow base for the game. The music never becomes stressful or intimidating. The initial idea was to add one track after each stage. Since there are only two stages in the game's current state, each adds three more tracks to reveal the complete composition eventually.

3.7.2 Sound Effects

Throughout the game, different types of sound effects add to the soundscape whenever the need arises. This can be the case when the player needs some indicator or notification concerning their abilities or environment. For instance, when they use one of their symptom abilities, and when it is ready to be used once more, sounds will play. The two sounds sound slightly different but resemble each other enough for the player to know they are referencing the same action. Further, hovering over and clicking buttons in user interfaces will provide the player with auditory feedback.
Overall, the audio used throughout the game has a subtle and soft sound, except for the sound effects used for the symptom abilities. As mentioned in 3.4, they strive to represent their specific symptom as well as possible. Thus, they are the most striking and disruptive sound heard throughout the game. First, a tinnitus is played whenever the player uses the Sensory Overload ability. The sound can be heard clearly at average volume; thus, it was not modified in any way. Second, a static noise accompanies the tinnitus noise during the ability. For the sake of gameplay, the sounds had to fulfill their purpose while simultaneously blending into the game well and, even though given their intrusive nature, not being too disruptive.

4 Development

The game accompanying this thesis was developed in Unreal Engine 5, primarily using the blueprint visual scripting system. This chapter explains the different steps of implementing the game's mechanics and other relevant gameplay elements. Further, it describes the creation process of the shaders used to enhance its overall optics. The following chapter does not represent a complete list of development steps but rather an overview of the most notable parts. It will reference some specific components and features the Unreal Engine provides. For detailed information, please refer to chapter 2.1 or the official documentation [79].

4.1 Mechanics

In this section the development of some notable parts of the playable character, including but not limited to movement, player vitals, combat, and further abilities, are elaborated on.

4.1.1 Player vitals

A blueprint component called *BPC_ PlayerStats* contains most of the health, stamina, and stress logic. The component is passed on to the *BP_ ThirdPersonCharacter* and the blueprints for training dummies and enemies. Since neither the enemies nor the dummies need any stamina or stress logic, the blueprint logic must differentiate between the various actor types. Upon creating the blueprint component, the blueprint owner is enquired, and a cast to the *BP_ ThirdPersonCharacter* is executed. If the cast fails, further casts to the *BP_ Dummy*, the *BP_ AI_ Fear*, and lastly, the *BP_ AI_ Anger* are called. When a cast succeeds, the character type is set in an Enumerator called *E_ CharacterTypes*. Essentially, the purpose of this query is to determine the kind of actor that is calling up this blueprint.
Afterward, the character‘s current health is transferred to the HUD‘s health bar by calling a *Set Percent* node and passing on the correct health value by dividing the current health by the maximum health. The HUD and the enemy‘s health bars are correctly passed on by connecting them to a *Select* node that chooses the correct item depending on the character type set in the enumerator. Also, using the same approach, the player‘s stress amount is transferred to the HUD. Further, *BPC_ PlayerStats* contains methods that decrease or increase health, stamina, or stress and increase the maximum value of health and stamina. When health, stamina, or stress fluctuation arises, the methods are called outside the blueprint

component. The initial health amount is set to 250, and the initial stamina amount is set to 200. Since the stress value will increase using symptom abilities, the initial value is 0 and can increase to a maximum of 100.

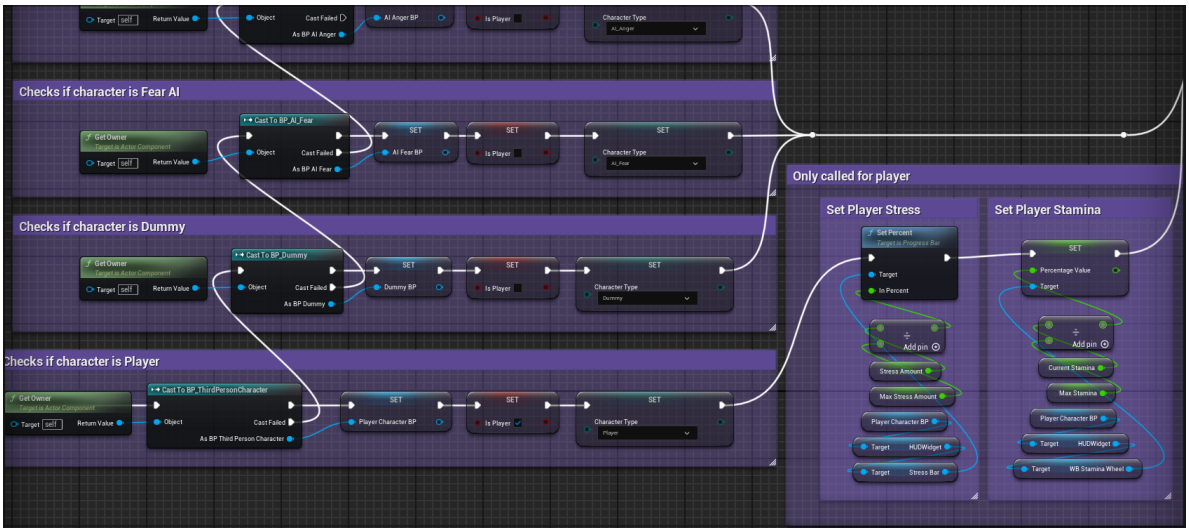

Figure 4.1: Applying the vitals of an actor in the *BPC_ PlayerStats* blueprint

4.1.2 Movement

Several basic movement abilities are already implemented in Unreal Engine 5 when starting a project with the third-person character template. These are walking, jumping, and looking around, for instance. In addition, more abilities were developed so the player could move through the levels. When setting up a new ability that the player is supposed to activate by pressing a key or using the mouse, a new *Input Action* needs to be created and then added to *Input Mapping Context* asset; a list that maps each action that the player can take to its respective key on a keyboard, a mouse, or a controller. The action can then be called in a blueprint, and the implementation of the ability can be connected to it. For example, figure 4.2 shows the sequence that will start once the key assigned to the vaulting action is pressed. With some exceptions, these calls are being made in the *BP_ ThirdPersonCharacter asset*. Further actions that were implemented by repeating the same process are as follows:

- *Sprinting* will increase the float *MaxWalkSpeed* that comes with the third-person template and will continuously drain stamina while the player presses the assigned key. Once they stop, the walking speed is returned to the previous value.
- By pressing the key assigned to the *dodging* action, the sequence will start by checking if the player has enough stamina to perform the action. Suppose this is true, and the player is not already dodging. In that case, any ongoing attacks are stopped, and a

Play Montage node plays a dodge animation in the character skeletal mesh component. Afterward, the stamina is depleted by 25.0.

- The player can *crouch* if they are not sprinting. The size of the player's capsule component will be decreased, and the camera's position will slightly move away.

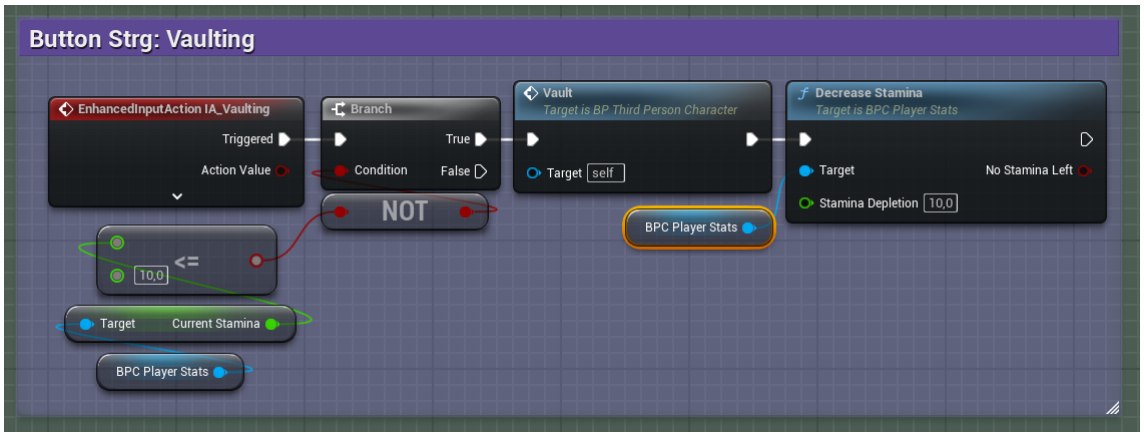

Figure 4.2: The vaulting action called by an assigned key

4.1.3 Combat

As a default weapon, the player uses a sword of two variations that are different in color and damage output. The swords are implemented through a blueprint called *BP_ Weapon*. When constructed, the blueprint receives information about the type of weapon from a data table called *DB_ Weapons*. The table gets its categories from a structured asset containing slots for name, damage, required level, icon, static mesh, type, and weapon socket. In the game's current state, the required level and type do not harbor any function, as there are no levels nor weapon types besides melee weapons. Leaving these in allows for potential expansion at a later point in time. The two swords were entered into the table separately to be based on different static meshes and carry out various amounts of damage. To increase the precision in combat, the player can lock their camera onto an enemy. Any eligible actor is returned by performing a sphere trace from the camera location to a point further down in its targeted direction and checking for any PhysicsBody or Pawn with the tag *damagable*. If this is the case, the actor is set to be the player's focus in the Event Tick until the player presses the assigned key again.

Symptom abilities

The symptom abilities act as an extension of the player's movement and combat skills. Their logic was implemented inside of the *BP_ ThirdPersonCharacter*. In individual methods, the

exact effects are defined, including activating any sound assets that play during the ability and activating a particular post-processing volume in the scene equipped with the concurrent visual effect. A post-process volume for each symptom is placed in each level where the game mode is set for the player to move around freely. The visuals supporting the ability are fine-tuned by adding effects provided by the post-process volume in the details panel or by adding post-processing materials that apply a specific visual effect.
One of the materials used for the *Derealization* ability can be seen in figure 4.3. The material is attached to the post-processing volume, harboring all effects of the derealization ability. To achieve distortion, a noise texture is added to the screen position of the currently rendered pixels and is then applied to distort the UVs of the scene texture. Besides applying the material, the global saturation levels of the post-processing volume are turned down significantly.

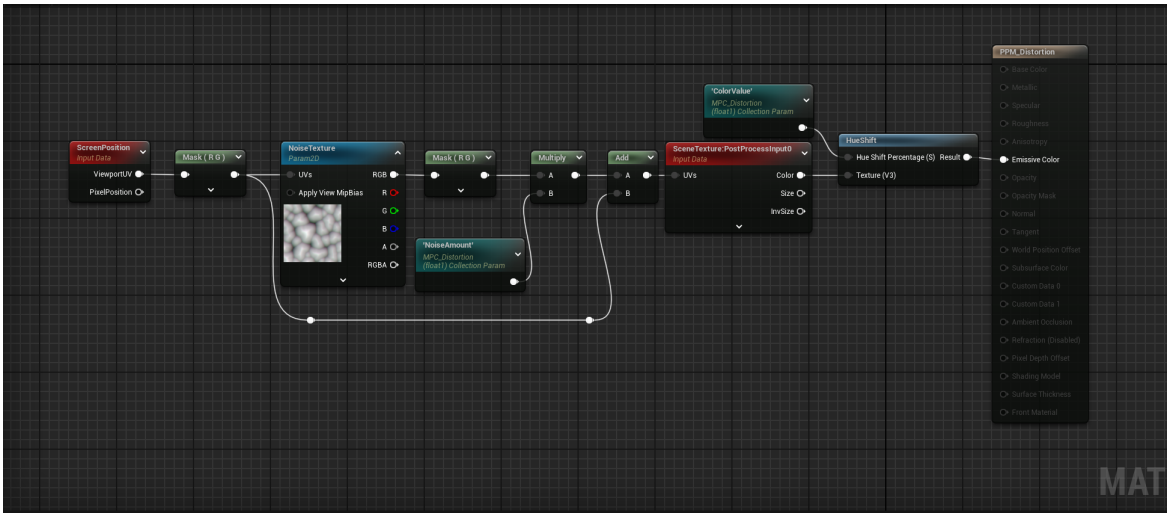

Figure 4.3: A post-processing material creating a distortion effect

Further, a Niagara component of all enemies in a radius of 5000 is activated by activating the derealization ability, playing a *fading away* effect. The Niagara effect *NS_ CharFadeAway* wraps itself around any character using the *SK_ Mannequin* skeletal mesh by adding the *Initialize Mesh Reproduction Sprite* to the Niagara node and selecting the skeletal mesh as the base. A high particle count is set to yield a smooth look to the particles. Due to the high count, the abovementioned radius limitation is applied to prevent performance from being compromised. It should be noted, though, that stuttering issues can arise on computers with lower performance.
The *Depersonalization* ability's visualization is implemented by adding a post-processing material that achieves a radial blur by applying *gaussian weights*. For that, a vector is drawn from the screen's center to the pixel currently being rendered. The middle of the screen is determined by subtracting a *TextureCoordinate* from a vector containing the values 0.5 and

0.5. Afterward, eleven values representing a Gaussian distribution are used as weights for a *Gaussian kernel*. Gaussian kernels are used for convolution with a Gaussian distribution to achieve a smoothing effect. The screen color is multiplied by these weights, added, and attached to a *Linear interpolation* expression along with the initial screen color. The process is inspired by a tutorial found at [80]. Similarly to the derealization ability, the post-processing material of the depersonalization ability decreases the global saturation. When the ability is enabled, the distance of the third-person camera is increased. Using a timeline that implements a cubic curve from the start distance value to the eventual distance value, the camera's movement gets smoothed.

The *Sensory Overload* ability activates a post-process volume whose temperature is set to a high value, and a black vignette is added. Its *Bloom* value is set to an increased intensity to achieve a blurry image as if the player had cloudy eyes. The exposure value is heavily increased to intensify all white spaces throughout the scene. A subtle *lens flare* effect is applied to simulate a glaring sun, and a motion blur mirrors the loss of the ability to process one's surroundings. The *film grain* property of the post-process volume is increased to 1, and the same radial blur post-processing material attached to the depersonalization ability is applied. Additionally to the visual effects, the tinnitus and static sounds increase in volume in the *Sensory Overload* function.

Specific changes in the player's vitals or movement are called from different blueprints. For instance, the temporary invincibility that the depersonalization ability grants is implemented in the *Decrease Health* function in the abovementioned *BPC_PlayerStats*. The method is called whenever the player gets hit by an enemy and starts by checking if the depersonalization ability is active via a boolean set in the depersonalization function itself. If it is true, the logic of health decrease is never called. The process is repeated in the *Decrease Stamina* function to prevent the player from consuming stamina while using the System Overload ability.
To turn off the effects properly following a set amount of time after enabling one of the abilities, a function called *Disable symptoms* provides all necessary actions to deactivate any of the abilities. Every Post Process Volume related to the abilities is equipped with the tag *symptom*, which the function looks up and disables. The volume of any auditory effects is set back to zero, and the enemies' visual *fading away* effect is disabled. All symptoms are activated by the same button and are differentiated by an int value *AttackSlot* that is changed by scrolling the mouse wheel. Depending on the int, one of the three abilities' functions is called. Afterward, a delay is called, setting the ability's duration. A second delay is called to set a cooldown timer, during which the player cannot use the ability again. A sound plays when activating the ability and when the cooldown is over.

4.2 Other gameplay elements

This section elaborates on various other elements found throughout the game that did not fit the description of the sections above. Nevertheless, they have a notable impact on the game and should be mentioned.

4.2.1 Artificial Intelligence

To facilitate a straightforward development of Artificial Intelligence, the engine provides several features that streamline the process. By combining these features, the enemy's behavior can be defined. The components used for this project will be explained in this section. During certain parts of the development, code found in [81] served as a guideline.
First, a *Blackboard* asset was created, which is used to store values relevant to the artificial intelligence in Unreal Engine. The *Blackboard* asset named *BD_AI* stores the values *SelfActor*, *TargetActor*, *inMeleeRange*, and *InvestigationLocation*, that can be set in the other artificial intelligence assets and retrieved by a *Behavior Tree* asset. The asset *BT_AI*, seen in figure 4.4, was set up. The *Blackboard* asset mentioned above was assigned to it, which means that the *Behavior Tree* asset can directly refer to the values of the *Blackboard* asset at any given point. The *Behavior Tree* instructs the enemy to either patrol around a set radius or chase and attack the player if they are detected. The precise logic of the AI detection will be revisited later in this section. The tree consists of different types of composites and is read from left to right, meaning the elements on the left will always have priority. If they are called under certain conditions being met, ongoing sequences with less priority will be halted immediately. Once a specific sequence starts, the procedures followed are defined by tasks, services, and decorators. Tasks are the core of the AI's behavior and will provide exact instructions on how to proceed. Services can be attached to tasks and other composites and are commonly used to update the Blackboard keys and control the flow of the *Behavior Tree* accordingly. Decorators attach conditions to their assigned composite that need to be met for it to trigger [82].

If the *Blackboard key TargetActor* is set, the condition for the *ChaseTarget* sequence is fulfilled, and the attached service *GetDistanceToTarget* will be triggered. The service receives the *Target key value* and compares the controlled pawn's distance to the float *MeleeAttackDistance*, which is set to 250 by default. Subsequently, the task *BT_Task_ChaseTarget* will be triggered, which utilizes the *AI MoveTo* node to move towards the target, i.e., the player. It will do so until a certain distance is reached, set to 100 by default. If the enemy is within this distance of the player, they will start the task *BT_Task_AttackTarget* where the method *Attack* is called upon, which is inherited by an interface assigned to all pawns with artificial intelligence logic implemented. The logic of the attack function is implemented in the blueprint of the enemy itself. It mirrors the player's attack process from the

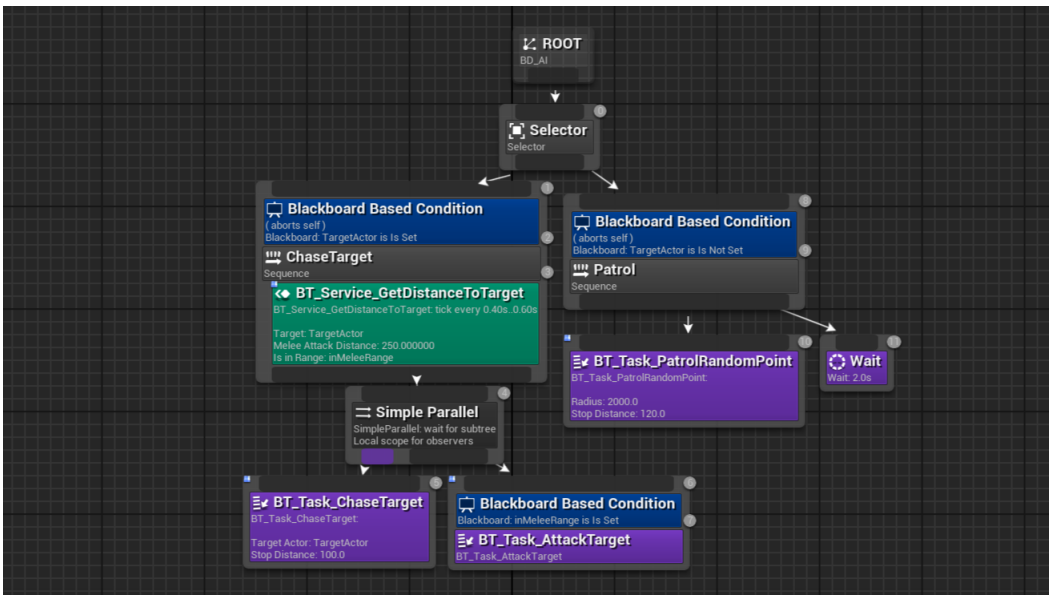

Figure 4.4: Control of the enemy‘s behavior using a *Behavior Tree* asset

BPC_AttackSystem with slight alterations to remove the player's input. If the player moves out of the *MeleeAttackDistance* or was never detected in the first place, the key *TargetActor* will not be set, and the condition for the *ChaseTarget* sequence will not be fulfilled any longer. The AI will maintain a second strand in the *Behavior Tree*. The patrol sequence will trigger whenever the key *TargetActor* is not set, meaning the player is not nearby. During the patrol sequence, the *BT_Task_PatrolRandomPoint* task will be first completed, followed by a wait time of two seconds. In the abovementioned task, the AI will again utilize the *AI MoveTo* node to move to a random location. The location it moves to can be 120 from where it first started, anywhere in a radius of 2000. Due to the priority setup, the patrol sequence will be interrupted immediately if the player enters the distance set in *MeleeAttackDistance*. The enemy blueprint is assigned the custom *AI Controller* class *BP_AI_Controller_Generic*. Following its *BeginPlay* node, the behavior tree is connected to the *Run Behavior Tree* node, which starts the process upon creation. Further, the controller implements a method that handles the player's detection, as shown in figure 4.5. The method receives two inputs: one actor variable and one *AIStimulus* variable. The *AIStimulus* is a predefined variable by the engine that can have different properties like hearing, damage, or sight. For this project, the enemy is merely supposed to receive sight stimuli. First, the incoming stimulus is compared to the type *AISense_Sight*. If it matches and is successfully sensed, the player is set as the target by accessing the *Blackboard* asset and setting the *Target Actor* key. If the player steps out of the bounds perceivable for the enemy, the value of the key will be set to *None* for the moment, and the enemy will start patrolling again.

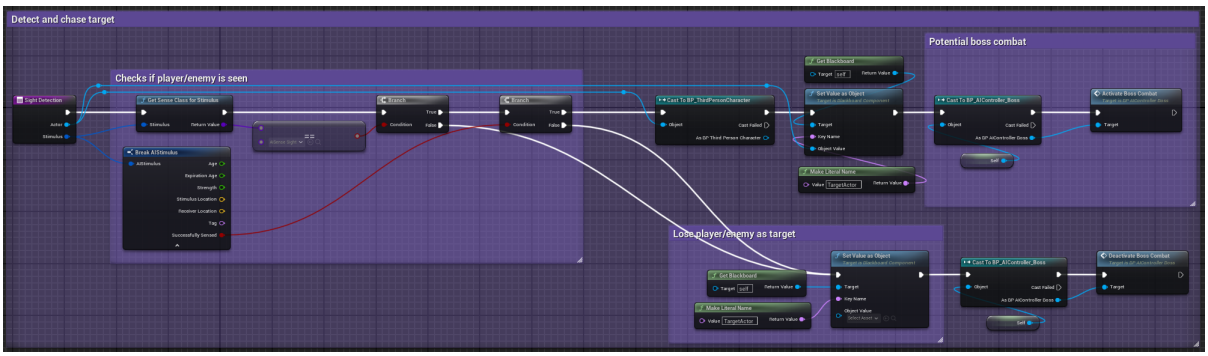

Figure 4.5: The sight detection method of the *BP_AI_Controller_Generic* blueprint

By default, the AI detection will receive stimuli from all pawns nearby, which leads to the problem of enemies attacking each other and having trouble detecting the player. To forego the problem, the DefaultGame.ini file was modified by adding the following lines:

```
[/Script/AIModule.AISense_Sight]
bAutoRegisterAllPawnsAsSources=false
```

The addition of this code excludes pawns, meaning enemies, from sending stimuli to each other.

4.2.2 Dialogue

To give the different characters personalities, a dialogue system was created that consists of different blueprints that house the dialogue itself and all the necessary functions to navigate through a dialogue tree. One blueprint class named *BP_NPC* combines the different functions. Different characters can be introduced to the game by creating children from this blueprint. Most of the dialogue system is handled within a *Blueprint Component* called *BPC_Dialogue_Base*. In order to move through a dialogue that takes the shape of a tree due to the different options of responding, a map with two integer values as indexes is created and used to determine the dialogue progression at any point. The following functions are crucial to the implementation of the dialogue system:

- *Open Dialogue* creates and adds the *Widget blueprint WB_Dialogue* to the viewport that displays the text and the possible responses the player can choose from. The method is called when a character is supposed to start talking and is triggered in the event graph at set points in the story or when the player interacts with them.
- *Add Dialogue* receives the name of the character, their dialogue, and all available responses and forwards them to the *Update Dialogue Text* function found in *WB_Dialogue*. The function applies the values to the texts on the user interface.

- *Update Selected Option* is called whenever a particular option is chosen and will add a new value to the map containing the dialogue progress.
- *Clear Dialogue Process* empties the map variable containing the dialogue progress.
- *Close conversation* removes the user interface from the viewport and resets the actor's interactibility.
- The *Dialogue* function is responsible for the structure of a conversation by connecting *Add Dialogue* functions with each other, filling them with the name of the character, the text, and the possible answers. The latter will then provide an index that leads to individual progressions in the dialogue tree, as shown in figure 4.6.

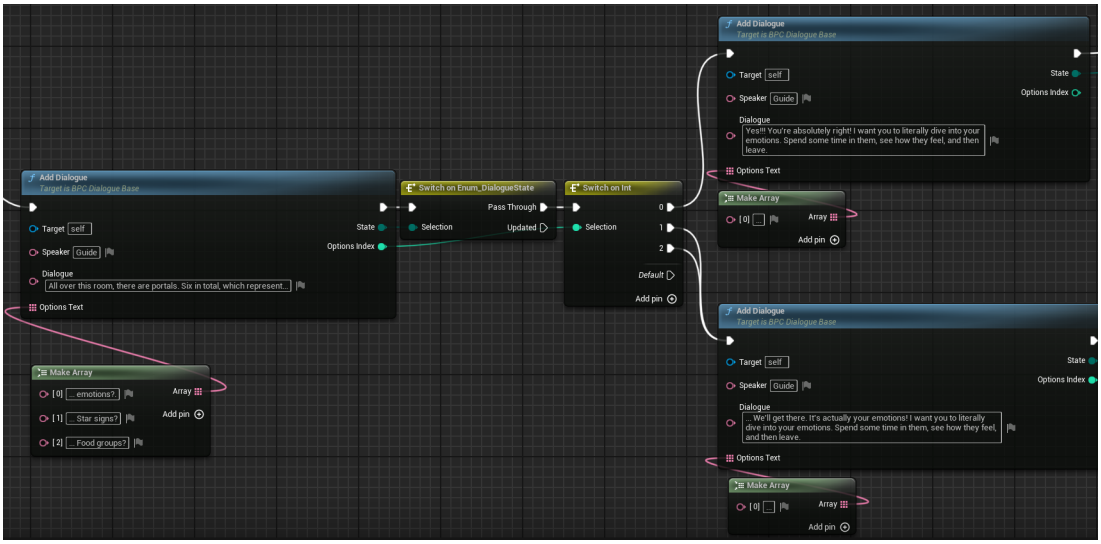

Figure 4.6: The *Dialogue* function setting up a conversation

4.2.3 User Interfaces

The user interfaces were created using widget blueprints explained in 2.1.1. In most cases, the elements needed for the particular display are added through the *UMG* interface and inserted in different horizontal and vertical boxes that group all objects they contain, regardless of screen size. This ensures the user interface will still look coherent on monitors of various resolutions. Buttons gain functionality by using a *On Clicked* expression, which is then connected to either a separate function or a direct action within the blueprint itself. The *head-up-display* mentioned in 3.6 is created and added to the viewport in the *Event Tick* of the *BP_ ThirdPersonCharacter* so that it always appears whenever the player character is enabled. Further, to enhance the coherence of the user interfaces, a custom cursor was assigned to the project by creating a widget blueprint that contains only one singular image

containing the cursor. The widget blueprint could be applied in the project settings under *Software Cursors*.

4.3 Shaders

The game used two shaders to achieve Moebius's overall art style.
Some of his main characteristics are clear, black outlines, so one shader was developed to apply outlines to all objects in a scene, seen in 4.7. The shader is based on a tutorial found in [83]. A material called *PP_ OutlineShader* was created, and the material domain *Post Process* was assigned to use it as such. The material is then set to apply its effect before tone mapping, which fixes issues with temporal anti-aliasing and GBuffer lookups and should be done when working with depth and normals [84]. To correctly identify the borders of all scene actors, the shader determines the difference in depth of the scene of each pixel's neighbor pixels. To achieve this, a material function called *MF_ GetKernel* is created that gets the position of every pixel and its neighbors in one frame and scales them depending on the screen's resolution. The function simultaneously outputs this information to another function called *MF_ MaskDepth*. The scene depth of the five pixels, which include the center pixel and its left, right, upper, and lower neighbors, gets retrieved, and the center pixel's depth is compared to the remaining pixels. The differences in depth are added up and compared to an overall threshold that can be modified at any given point. In the material function *MF_ DetectEdges_ Depth*, a value of 1 is returned if the value is above the threshold, and the pixel is considered on an edge and will later receive an outline. The idea behind this is that the difference in scene depth is minimal on a surface but very high on object edges. This procedure is equivalent to a convolution with an edge-detection kernel on a depth image.

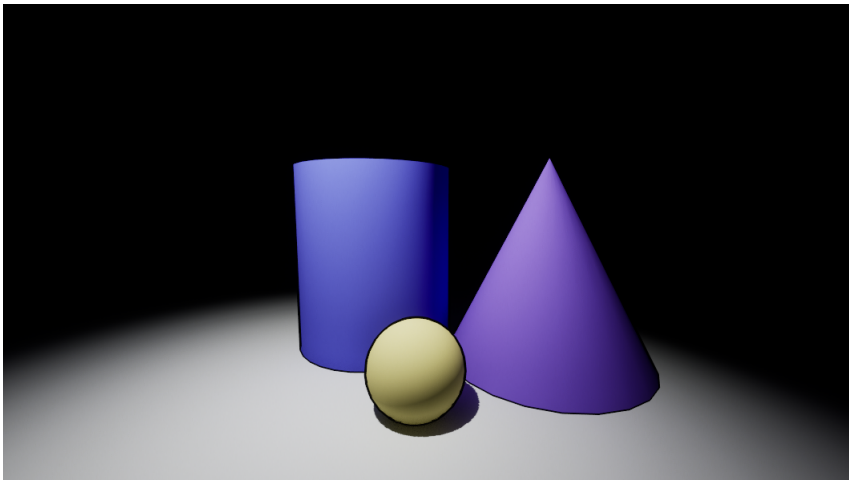

Figure 4.7: The outline shader demonstrated on simple shapes

All the functions mentioned above are called inside of *PP_ OutlineShader* and are provided with parameters. The function implementing the edge detection is inserted into the alpha channel of a *linear interpolation* node. Further, functions containing the logic concerning the amount of drawing distance and overall line quality improvement are inserted into the *A* slot, and a color parameter is inserted into the *B* slot. Besides the logic that enhances the overall quality of the lines, a *SceneTexture* node is applied to the A slot of the Linear interpolation so that the pixels of the unmodified scene are shown whenever the alpha value is 0. Due to the function returning 1 as a value, the linear interpolation will take this as an alpha value whenever an edge is detected and color it white. On top of this pixel, the shader will add the parameter‘s color and the logic for this pixel only. To swiftly modify the parameters set in the material later, the material instance *PPI_ OutlineShader* was created.

A second characteristic of Moebius‘ illustrations is the use of flat colors throughout most scene elements. Thus, another shader was developed based on a tutorial provided under [85]. The shader aims to achieve the *Flat color* style, seen in 4.8. The post-processing material that manages this shader is called *M_ FlatColor*. To isolate the lighting information, two *SceneTexture* nodes, one containing the color information of the scene stored in the *G buffer*, and one containing information about the base color, are used. Both outputs are desaturated to turn them into gray-scale values. The two values are divided and applied to the *A* slot of an *if* expression. The expression compares it to a second value and returns 1 if the first value and 0 if the latter is bigger. Essentially, this means that the expression returns shadows and light in stark contrast. The previous steps are then used to mask the scene‘s shadows. The mask is used as the alpha channel of a *LinearInterpolate* expression, which receives a linear interpolation between the two *SceneTexture* expressions and a shadow color assigned by the user. Keeping in mind that the desaturated shadow and light values are acting as a mask, if the *LinearInterpolate* expression connected to the material output returns a 1, meaning the current pixel represents a light value, the linear interpolation of the two *SceneTexture* expressions will return its color value. If it returns a 0, the shadow color will be assigned. The scene‘s colors will then return a matte color with hard shadows. Utilizing multiple *SceneDepth* expressions, the drawing distance is set to exclude far-away objects from the calculations. Additionally, a texture sample will determine the pattern of the shadows. The blueprint can be viewed by accessing the file *FlatColor_ Material.PNG* attached to this thesis on the flash drive.

The shaders are attached to an existing Post Process Volume that is then set to apply the effect to an infinite extent throughout the scene. To harmonize with the outline shader, all meshes must have as few edges and lines as possible. Due to the shader drawing lines at any borders that can be located, meshes potentially lose details when too many outlines are applied to them. To forego this, in the material that applies a *Normal map* to an actor, they were connected to a *Substract node* and, thus, lost some of their intensity. This results in

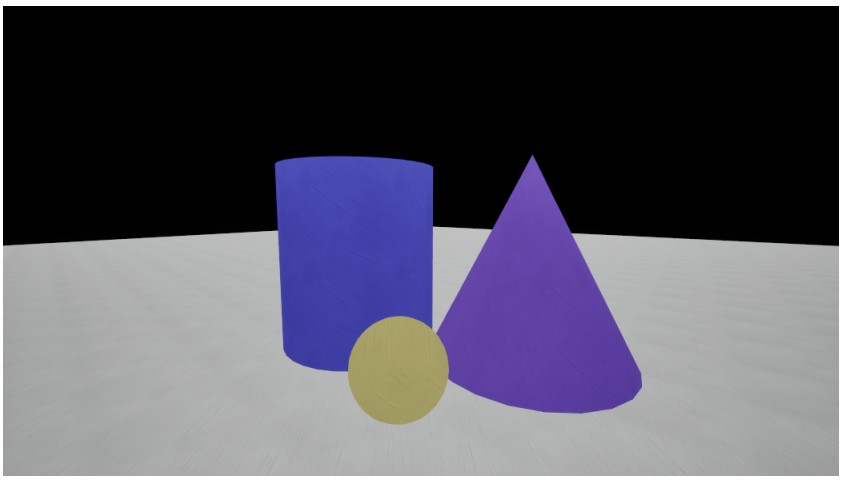

Figure 4.8: The flat color shader demonstrated on simple shapes

the particular objects still exhibiting strong outlines around their borders but only hinting at details on their surface, which can be seen in figure 4.9.

By default, the shaders will alter every actor in the scene. However, not every actor will benefit from the shader's effect due to its size or nature. For instance, the latter applies whenever user widgets are displayed in the world space. In order to prevent lines from being drawn on the border of an object, the *Render CustomDepth Pass* option needs to be checked, and a *Custom Depth Stencil Value* of 1 needs to be set. Activating the *CustomDepth Stencil* will enable certain actors to be left out of the calculations concerning a shader. A separate *SceneTexture* expression is connected to the material output that handles how the specific actor should be calculated instead.

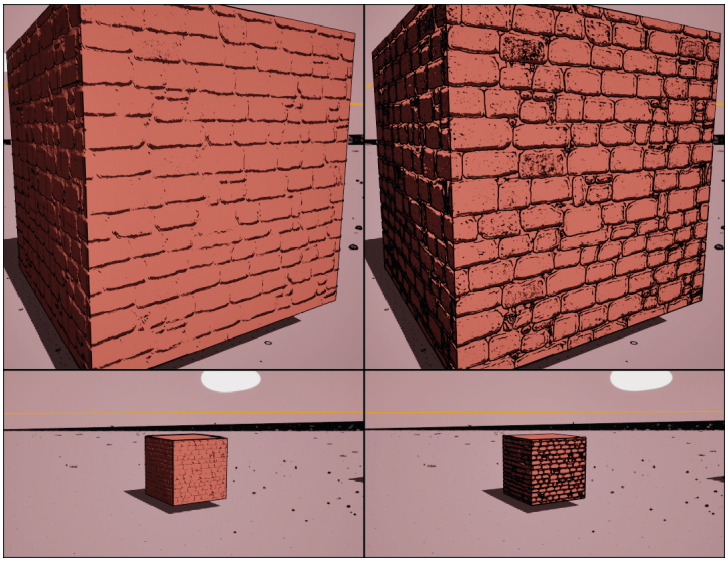

Figure 4.9: Object on the right losing details due to too many outlines

For a better visualization of the effects applied, the difference between the scene with and without the two shaders is illustrated in figure 4.10.

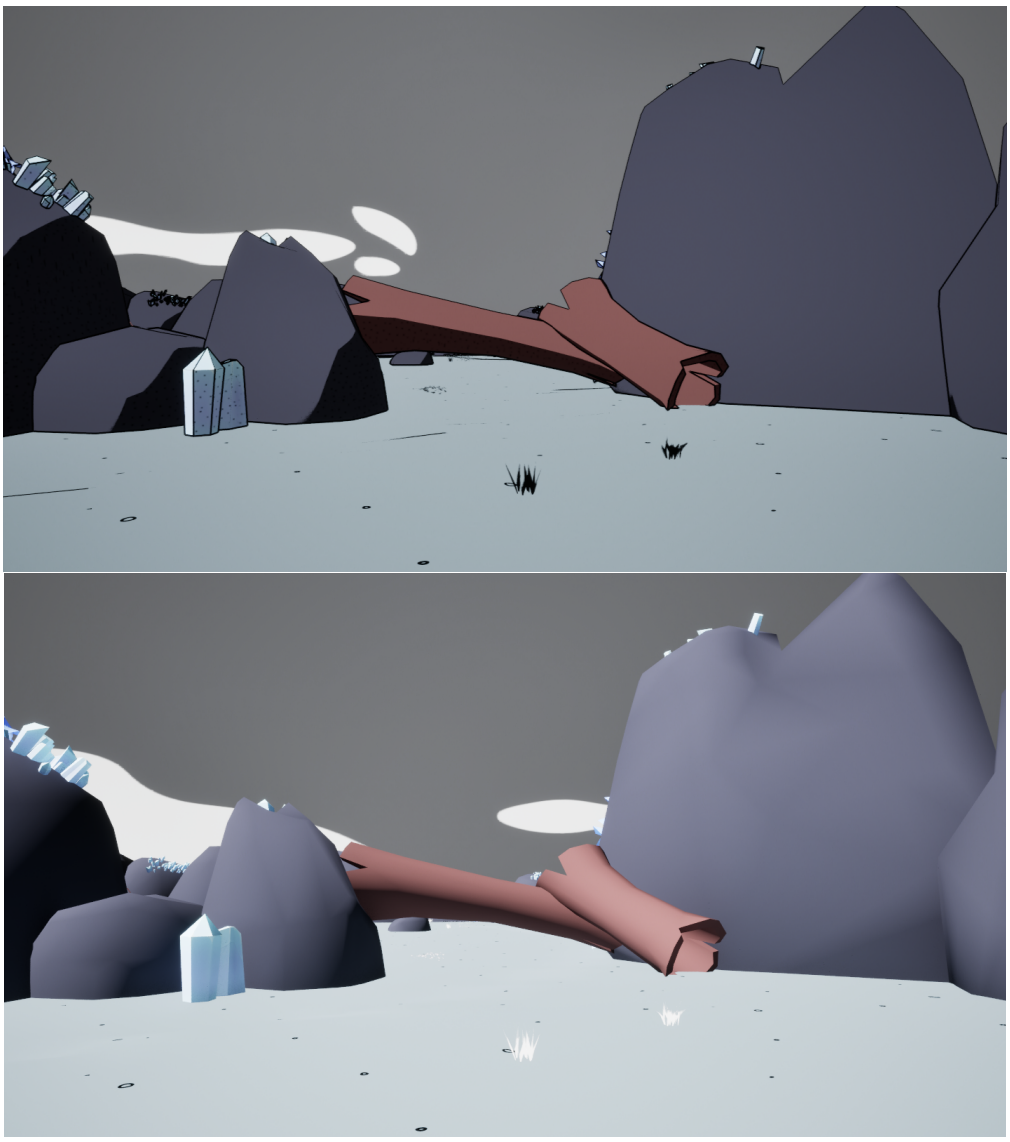

Figure 4.10: Scene with shaders (Top image) and without shaders (Bottom image)

5 Conclusions and outlook

For this thesis, the representation of mental illness in video games and its impact on society was presented based on existing video games and scientific findings. Examples of strategies for achieving a positive representation were analyzed and discussed. Based on these results, a video game prototype was created to attempt a combination of an accurate and layered portrayal of generalized anxiety disorder and a gameplay experience that is generally enjoyable. The aim was to destigmatize mental illness and to relay information about the topic, which is not necessarily easy to grasp for people who have never had any points of contact with it.

Eye of the Storm leads the player through two stages representing the protagonist's emotions and a hub called *the void*, which acts as a metaphor for the character's inner emptiness. The void is slowly filled with the environment of each stage they visit, thus hinting at the protagonist letting their emotions in. A character called *the guide* leads the player through the process and offers an exposition on anxiety. The game introduces three symptoms of generalized anxiety disorder and utilizes them as abilities that help the player in combat. In order to prevent romanticizing them, a stress bar fills up whenever they are used. When the stress bar peaks, the player will experience a shutdown and cannot attack for a short while or take more damage. By using the Unreal Engine the development significantly benefited by contributing certain features like the template for the Third-person character. The template streamlined the implementation of mechanics to the player character by already inheriting the basic movement and functions needed for a playable character.

5.1 Goals met

In addition to depicting anxiety through levels, environment, and abilities, the game characters welcomed the topic but did not romanticize it. The protagonist faced feelings and experienced their inner processes head-on, meeting the criteria of implementing a eudaimonic approach to the topic.

The guide's character was designed to relay the needed information in a playful way that does not feel too much of a burden and to add some lightheartedness to the topic. Adding this character was a good choice from the developer's perspective, as the game became more well-rounded and lively afterward. The character appears whenever the player requires directions or explanations about the situation at hand. First playtesters offered positive remarks

about the guide and the overall dialogue and noticed that the character added warmth and guidance to the game. In some cases, the dialogue was named the part that stood out the most to them.
Further, after adding the symptom abilities and balancing their values, the combat became more dynamic rather than linear. The different abilities each show a clear purpose and have a justification to be implemented, which might lead to the players not focusing on one particular ability but using all three abilities during combat. The balance between gameplay and the visualization of symptoms was met, as the symptom abilities never became too disruptive.
This project adds to the conversation by introducing a somewhat fresh approach to the topic. While many games thematize mental health, this project is a more specific approach to generalized anxiety disorder in particular. Further, the action-adventure genre lacks the representation of mental health that other genres, like sidescrollers, receive. Resultingly, a significant addition to the representation of mental illness in video games was made by this thesis and the accompanying game, *Eye of the Storm*.
Adding the shaders and the accompanying graphical art style based on the artist *Moebius'* illustrations achieved the intended effect and added to the dream-like fantasy setting of the world the player finds themselves in.
Initially, one playable level was planned to provide an overview of the game's concept. However, in order to convey the differences in level structure depending on an emotion, a second level was added. The contrast between the two is stark and brings the desired effect. The feedback of the first playtesters suggests that the balance between painting anxiety as something natural and acknowledging its impact on people experiencing it has been achieved.

5.2 Challenges and limitations

Initially, the plan was to develop a game set from a first-person perspective to visualize the symptoms more vividly. After the first tests of the shaders and symptom abilities, a substantial lack of orientation could be observed, and thus, an approach with the third-person perspective was implemented successfully. Further, a more comprehensive range of characters was planned to fill the game with life. Due to time constraints, one character was decided upon. However, the guide fits the game's overall theme and provides a large amount of warmth, as the first game testers noted. A significant limitation was the lack of expert opinions due to the project's scope and time constraints. As mentioned in 1.1.2, an extensive amount of feedback and personal experience was provided during the development of *Hellblade: Senua's sacrifice* both by people experiencing the illness and mental health professionals alike. These points of view resulted in a depiction of Schizophrenia that is considered to be both accurate and free of any evaluation. Given more time and access to voices relevant to the topic, *Eye of the Storm* would have significantly benefited from these

contributions and the feedback concerning specific visual and auditory imitations and the delivery of this sensible topic to the player. This is especially important, as the accuracy of such visual and auditory imitations are disputed, as mentioned in 1.1.3. Further, some minor bugs concerning the symptom abilities and the shutdown they induce made playtesting difficult for an extended time. Ultimately, the issues could be resolved, and playtesting could continue, but essential time was lost.

5.3 Future work on the project

A future game iteration should add further stages to complete the emotion wheel mentioned in 3.2.2. These stages would again be displayed in color schemes associated with the particular emotion. Further, the character models of the guide, the enemies, and perhaps most importantly, the protagonist could be equipped with character models chosen on a common theme to deploy significant coherence throughout the game. Initially, the different characters were supposed to resemble humanoid animal characters. Presumably, this would have also added to the dream-like fantasy setting of the game. A further feature touched on early in development was adding non-playable characters throughout the level that the player could help with along the way. These characters would have their own personalities and be metaphors for secondary emotions tied to the emotion of their level. Secondary emotions often appear after the primary emotion has been experienced. An extract from these secondary emotions can be seen in figure 3.3. After meeting the character, the player would be asked to help them in some form. The different stages would act more like isolated storylines with several quests that would require the player to stay in them longer. Additionally, the game has a great potential to build on the role-play element it currently hints at. Several functions, like weapon types and magic projectiles, were developed but did not add enough value or show any purpose to the game to be implemented.

5.4 Outlook

The combination of gameplay and the topic of mental health will likely be taken up increasingly in the years to come. While researching the topic, an extensive list of games that already strive to do so was gathered. The games mentioned in 1.1.2 only scraped the surface and were carefully chosen from a long list because of their specific value to this project. Developers should consider the impact their game can have on society when dealing with mental health topics. For the future of action-adventure games, it should be noted that the trade-off between creating a fun experience and providing founded information about topics like mental illness is critical to attracting more people, especially younger audiences.

6 Acknowledgements

I would like to extend special thanks to Fabio Anthony, who not only supported me in writing this thesis and bringing the project to life but also encouraged me to pursue my aspiration of developing games. Further, I thank my parents for making my studies possible and helping me in all aspects of life. Thank you to everyone who supported me in writing this thesis, whether it was by providing feedback or reassuring words. Special thanks go to Michael Weiler who provided me with a beautiful score for the different stages.

Appendix

Along with this thesis, the following files are enclosed on a flash drive due to their relevancy to the project.

Appendix A: Game Prototype

The prototype of the game as an executable, along with all files necessary for it to work, is found on the flash drive in a folder called *Executable*.

Appendix B: Unreal Engine 5 project

A copy of the project as well as all required files, are stored on the flash drive in a folder called *Eye of the Storm*.

Appendix C: Soundtrack

Further, the complete version of the game's soundtrack, written and recorded by Michael Weiler, is stored on the flash drive under the name *EyeOfTheStorm_Soundtrack.wav*.

Appendix D: Asset list

A list of all external assets used in the game can be found on the flash drive in the folder *Documentation* under the name *AssetList_EyeOfTheStorm.pdf*.

Appendix E: Game Dialogue

Lastly, an excerpt of the conversation between the guide and the protagonist with all possible responses is saved on the flash drive in the folder *Documentation* under *GameplayElements_Dialogue.pdf*.

List of Figures

List of Tables

Bibliography

[1] Nicolas Rüsch, Matthias C. Angermeyer, and Patrick W. Corrigan. "Mental Illness Stigma: Concepts, Consequences, and Initiatives to Reduce Stigma". In: *European Psychiatry* 20.8 (2005), pp. 529–539. DOI: 10.1016/j.eurpsy.2005.04.004.

[2] *Fact Sheet: Common Mental Health Myths and Misconceptions*. Mental Health Commission of Canada. Apr. 28, 2023. URL: https://mentalhealthcommission.ca/resource/fact-sheet-common-mental-health-myths-and-misconceptions/ (visited on 06/24/2024).

[3] Diana Rose et al. "250 Labels Used to Stigmatise People with Mental Illness". In: *BMC health services research* 7 (2007), p. 97. DOI: 10.1186/1472-6963-7-97.

[4] Jai Ahuja and Palak A. Fichadia. "Concerns Regarding the Glorification of Mental Illness on Social Media". In: *Cureus* 16 (2024). DOI: 10.7759/cureus.56631.

[5] Leonard Reinecke and Mary Beth Oliver. "Media Use and Well-Being: Status Quo and Open Questions". In: *The Routledge Handbook of Media Use and Well-Being: International Perspectives on Theory and Research on Positive Media Effects*. 2016, pp. 3–13.

[6] Majid Bidaki et al. "Evaluating the Effects of Experiencing Virtual Reality Simulation of Psychosis on Mental Illness Stigma, Empathy, and Knowledge in Medical Students". In: *Frontiers in Psychiatry* 13 (2022). DOI: 10.3389/fpsyt.2022.880331.

[7] Manuela Ferrari et al. "Gaming With Stigma: Analysis of Messages About Mental Illnesses in Video Games". In: *JMIR Mental Health* 6 (2019). DOI: 10.2196/12418.

[8] Jozef Buday et al. "Depiction of Mental Illness and Psychiatry in Popular Video Games Over the Last 20 Years". In: *Frontiers in Psychiatry* 13 (2022). DOI: 10.3389/fpsyt.2022.967992.

[9] Sky LaRell Anderson. "Portraying Mental Illness in Video Games: Exploratory Case Studies for Improving Interactive Depictions". In: *The Journal of the Canadian Game Studies Association* 13.21 (2020), pp. 20–33. DOI: 10.7202/1071449ar.

[10] *Hellblade: Senua's Sacrifice Store Page*. Valve Corporation. URL: https://store.steampowered.com/app/414340/Hellblade_Senuas_Sacrifice (visited on 06/04/2024).

[11] Jodie Austin. "'The Hardest Battles Are Fought in the Mind': Representations of Mental Illness in Ninja Theory's Hellblade: Senua's Sacrifice". In: *The International Journal of Computer Games Research* 21 (4 2021). ISSN: 1604-7982.

[12] Ninja Theory. *Hellblade: Senua's Sacrifice*. Aug. 8, 2017.

[13] *Hellblade: Senua's Psychosis | Mental Health Feature*. Ninja Theory. URL: https://www.youtube.com/watch?v=31PbCTS4Sq4 (visited on 06/02/2024).

[14] *What Remains of Edith Finch*. Valve Corporation. URL: https://store.steampowered.com/app/501300/What_Remains_of_Edith_Finch/ (visited on 06/02/2024).

[15] Giant Sparrow. *What Remains of Edith Finch*. Apr. 24, 2017.

[16] David Silbert. *5 packende Geschichten aus What Remains of Edith Finch, die euch nicht mehr loslassen*. URL: https://store.epicgames.com/de/news/5-haunting-tales-from-what-remains-of-edith-finch-that-sit-with-you (visited on 08/14/2023).

[17] *Gris Store Page*. Valve Corporation. URL: https://store.steampowered.com/app/683320/GRIS/ (visited on 06/02/2024).

[18] Devolver Digital Nomada Studio. *Gris*. Dec. 13, 2018.

[19] *Night in the Woods Store Page*. Valve Corporation. URL: https://store.steampowered.com/app/481510/Night_in_the_Woods/ (visited on 06/04/2024).

[20] Secret Lab Infinite Fall. *Night in the Woods*. Feb. 21, 2017.

[21] Kathrin Trattner. "Night in the Woods : a Review". In: *Gamevironments* (6 2017), pp. 130–137. URL: https://www.researchgate.net/publication/317397592_Night_in_the_Woods_a_Review (visited on 06/04/2024).

[22] Heather Wald. *Night in the Woods' Portrayal of Mental Health Reminded Me that it's Okay to not be Okay*. Gamesradar. 2018. URL: https://www.gamesradar.com/night-in-the-woods-portrayal-of-mental-health-reminded-me-that-its-okay-to-not-be-okay/ (visited on 06/04/2024).

[23] Ellie Kovach. *The Real Horror of Night in the Woods*. The Believer. Apr. 5, 2022. URL: https://www.thebeliever.net/logger/the-real-horror-of-night-in-the-woods/ (visited on 06/03/2024).

[24] Nicky Case. *Nicky Case: itch.io profile*. itch.io. URL: https://ncase.itch.io/ (visited on 06/06/2024).

[25] Nicky Case. *Adventures With Anxiety*. 2019.

[26] Nicky Case. *Adventures with Anxiety: Screenshots*. URL: https://ncase.me/anxiety/sharing/ (visited on 06/06/2024).

[27] Living Well with Schizophrenia. *Are Schizophrenia Simulations Accurate?* May 27, 2020. URL: https://www.youtube.com/watch?v=60n7ZP3Cj-4&ab_channel=LivingWellwithSchizophrenia (visited on 06/25/2024).

[28] Vanmeijer. *Realistic Schizophrenic Simulation.* Mar. 8, 2022. URL: https://www.youtube.com/watch?v=V_RdxI3ZiPs&ab_channel=Vanmeijer (visited on 06/04/2024).

[29] Mike Thomsen. *History of the Unreal Engine.* IGN. Feb. 24, 2010. URL: https://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine (visited on 05/19/2024).

[30] Daniel Kayser. *Exploring the Mindset Behind Hellblade: Senua's Sacrifice.* Epic Games. Aug. 8, 2017. URL: https://www.unrealengine.com/es-ES/developer-interviews/exploring-the-mindset-behind-hellblade-senuas-sacrifice (visited on 06/26/2024).

[31] Jeff Farris. *Forging New Paths for Filmmakers on "The Mandalorian".* Epic Games. Feb. 20, 2020. URL: https://www.unrealengine.com/fr/blog/forging-new-paths-for-filmmakers-on-the-mandalorian (visited on 06/26/2024).

[32] *Licensing.* Epic Games. URL: https://www.unrealengine.com/en-US/license (visited on 06/26/2024).

[33] *Introduction to Blueprints - Unreal Engine 4.27 Documentation.* Epic Games. URL: https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/ (visited on 05/19/2024).

[34] *Materials - Unreal Engine 4.27 Documentation.* Epic Games. URL: https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/Materials/ (visited on 06/24/2024).

[35] *Material Expressions Reference - Unreal Engine 5.0 Documentation.* Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-material-expressions-reference?application_version=5.0 (visited on 06/25/2024).

[36] *Post Process Materials - Unreal Engine 5.4 Documentation.* Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/post-process-materials-in-unreal-engine (visited on 05/19/2024).

[37] *Post Process Effects.* Epic Games - Unreal Engine 5.4 Documentation. URL: https://dev.epicgames.com/documentation/de-de/unreal-engine/post-process-effects-in-unreal-engine (visited on 05/19/2024).

[38] *Niagara Overview - Unreal Engine 5.4 Documentation.* Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/overview-of-niagara-effects-for-unreal-engine?application_version=5.4 (visited on 06/26/2024).

[39] *Third Person Template - Unreal Engine 5.3 Documentation*. Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/third-person-template-in-unreal-engine?application_version=5.3 (visited on 06/26/2024).

[40] *Creating and Displaying UI - Unreal Engine 4.27 Documentation*. Epic Games. URL: https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/UMG/HowTo/CreatingWidgets/ (visited on 06/26/2024).

[41] *Sequencer Overview*. Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-sequencer-movie-tool-overview (visited on 06/26/2024).

[42] *Unreal Emgine Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/store (visited on 06/26/2024).

[43] *Shader*. World of VR. URL: https://worldofvr.de/shader/ (visited on 06/23/2024).

[44] *What is vector art?* Adobe. URL: https://www.adobe.com/creativecloud/illustration/discover/vector-art.html (visited on 06/26/2024).

[45] *Anxiety: Signs and Treatment Options for Anxiety Disorder*. Stanford Center for Health Education, June 13, 2022. URL: https://www.youtube.com/watch?v=wHZV7xvt49A (visited on 05/19/2024).

[46] Philip R. Muskin. *What are Anxiety Disorders*. American Psychiatric Association. June 1, 2023. URL: https://www.psychiatry.org/patients-families/anxiety-disorders/what-are-anxiety-disorders (visited on 05/26/2024).

[47] Svenn Torgersen. "Genetic Factors in Anxiety Disorders". In: *Archives of General Psychiatry* 40.10 (1983), pp. 1085–1089. DOI: 10.1001/archpsyc.1983.01790090047007.

[48] *Anxiety - Reversing the Vicious Cycle*. Government of Western Australia: Department of Health. URL: https://www.healthywa.wa.gov.au/Articles/A_E/Anxiety-reversing-the-vicious-cycle (visited on 06/23/2024).

[49] Richa Shri. "Anxiety: Causes and Management". In: *The Journal of Behavioral Science* 5 (2009), pp. 100–118. URL: https://www.researchgate.net/publication/264877930_Anxiety_Causes_and_Managements (visited on 05/26/2024).

[50] *Panic Disorder: When Fear Overwhelms*. National Institute of Mental Health. URL: https://www.nimh.nih.gov/health/publications/panic-disorder-when-fear-overwhelms (visited on 05/19/2024).

[51] *Panic Attacks and Panic Disorder*. Cleveland Clinic. URL: https://my.clevelandclinic.org/health/diseases/4451-panic-attack-panic-disorder (visited on 05/19/2024).

[52] *Types of Anxiety Disorders*. Mental Health UK. URL: https://mentalhealth-uk.org/help-and-information/conditions/anxiety-disorders/types/ (visited on 05/19/2024).

[53] *Generalized Anxiety Disorder (GAD)*. John Hopkins Medicine. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/generalized-anxiety-disorder (visited on 05/26/2024).

[54] Jitender Sareen Murray B. Stein. "Generalized Anxiety Disorder". In: *The New England Journal of Medicine* 373 (2015), pp. 2059–2068. DOI: 10.1056/NEJMcp1502514.

[55] *The ICD-10 Classification of Mental and Behavioural Disorders*. 2024. ISBN: 9789241544559.

[56] Micah Abraham. *Derealization - A Scary Anxiety Symptom*. CalmClinic. Jan. 17, 2023. URL: https://www.calmclinic.com/anxiety/symptoms/derealization (visited on 06/23/2024).

[57] Nicole Arzt. *Derealization Vs. Depersonalization: Understanding the Differences*. ChoosingTherapy. Apr. 4, 2023. URL: https://www.choosingtherapy.com/derealization-vs-depersonalization/ (visited on 06/23/2024).

[58] *Sensory Overload is Real and Can Affect Any Combination of the Body's Five Senses: Learn Ways to Deal With It*. Queensland Government: Queensland Health. July 15, 2022. URL: https://www.health.qld.gov.au/newsroom/features/sensory-overload-is-real-and-can-affect-any-combination-of-the-bodys-five-senses-learn-ways-to-deal-with-it (visited on 06/23/2024).

[59] Hailey Shafir. *Feelings of Emptiness: Causes, Getting Help, and Ways to Cope*. Apr. 29, 2022. URL: https://www.choosingtherapy.com/feelings-of-emptiness/ (visited on 06/23/2024).

[60] *Generalized Anxiety Disorder: Learn More - Treatment Options for Generalized Anxiety Disorder*. Institute for Quality and Efficiency in Health Care. July 7, 2021. URL: https://www.ncbi.nlm.nih.gov/books/NBK279594/ (visited on 05/26/2024).

[61] *Action-Adventure Game*. Video Game History Wiki. URL: https://videogamehistory.fandom.com/wiki/Action-adventure_game (visited on 06/25/2024).

[62] Michael Filimowicz. *Layouts*. Medium. 2023. URL: https://medium.com/understanding-games/layouts-913b8384c257 (visited on 06/10/2024).

[63] *What is a Hub World?* How to Make an RPG. 2017. URL: https://howtomakeanrpg.com/a/what-is-a-hub-world.html (visited on 06/10/2024).

[64] *POLYDesert - Asset Store*. Unity. URL: https://assetstore.unity.com/packages/3d/environments/landscapes/polydesert-107196 (visited on 06/25/2024).

[65] *Infinity Blade: Grass Lands in Epic Content - UE Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/product/infinity-blade-plain-lands (visited on 06/25/2024).

[66] *Lowpoly Handpainted Environment in Environments - UE Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/product/hand-painted-environment (visited on 06/25/2024).

[67] *LUSH: Stylized Environment Set in Environments - UE Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/product/lush-stylized-environment-set (visited on 06/25/2024).

[68] Christina Shore. *The Power of the Emotional Wheel: How to Use It for Greater Self-Awareness and Emotional Intelligence*. June 2, 2023. URL: https://besttherapies.org/the-power-of-the-emotional-wheel-how-to-use-it-for-greater-self-awareness-and-emotional-intelligence/ (visited on 06/22/2024).

[69] Imi Lo. *Structural Dissociation and Hiding Behind "Normal"*. Psychology Today. Nov. 26, 2023. URL: https://www.psychologytoday.com/us/blog/living-with-emotional-intensity/202311/structural-dissociation-and-hiding-behind-normal#:~:text=Trauma%20splitting%20involves%20a%20division,and%20Schema%20Therapy%20may%20help. (visited on 06/25/2024).

[70] *Forest Creature 2 in Characters - UE Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/product/forest-creature-2 (visited on 06/25/2024).

[71] *Golem in Characters - UE Marketplace*. Epic Games. URL: https://www.unrealengine.com/marketplace/en-US/product/golem-10 (visited on 06/25/2024).

[72] Jeffrey Ditzell. *Dissociation as a Defense Mechanism: Why It Happens and How to Handle It*. Healthline. Nov. 7, 2022. URL: https://www.healthline.com/health/mental-health/dissociative-defense-mechanism#as-a-defense-mechanism (visited on 06/23/2024).

[73] Arjan Terpstra. *Artist Spotlight: Moebius | Cook and Becker*. May 3, 2022. URL: https://www.cookandbecker.com/en/article/346/artist-spotlight-moebius.html (visited on 06/12/2024).

[74] *Ligne Claire*. Fandom, Inc. URL: https://aesthetics.fandom.com/wiki/Ligne_Claire (visited on 06/24/2024).

[75] *Moebius Gives 18 Wisdom-Filled Tips to Aspiring Artists (1996)*. Open Culture. Mar. 4, 2015. URL: https://www.openculture.com/2015/03/moebius-gives-18-wisdom-filled-tips-to-aspiring-artists-1996.html (visited on 06/12/2024).

[76] Kendra Cherry. *The Color Blue: Meaning and Color Psychology.* Verywellmind. Feb. 20, 2024. URL: https://www.verywellmind.com/the-color-psychology-of-blue-2795815 (visited on 06/24/2024).

[77] Aliza Ackerman. *A Guide to Color Meaning.* Adobe. URL: https://www.adobe.com/creativecloud/design/discover/color-meaning.html (visited on 06/24/2024).

[78] Behnam Mehrafrooz. *5 Golden Rules of Game HUD Design: Show, Don't Tell!* Pixune Studios. May 2, 2024. URL: https://pixune.com/blog/game-hud-design/ (visited on 06/19/2024).

[79] *Unreal Engine 5.4 Documentation.* Epic Games. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-4-documentation (visited on 06/25/2024).

[80] Ben Cloward. *Radial Blur Post Processing - Shader Graph Basics - Episode 55.* July 28, 2022. URL: https://www.youtube.com/watch?v=KLOGaHgo2GY&t=974s (visited on 06/26/2024).

[81] Gorka Games. *Unreal Engine 5 RPG Tutorial Series.* Feb. 20, 2023. URL: https://www.youtube.com/playlist?list=PLiSlOaRBfgkcPAhYpGps16PT_9f28amXi (visited on 06/17/2024).

[82] *Artificial Intelligence - Unreal Engine 4.27 Documentation.* Epic Games. URL: https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/ (visited on 06/19/2024).

[83] Evans Bohl. *How to Make Toon Outlines in UE5.* Aug. 6, 2022. URL: https://www.youtube.com/watch?v=9KvUfnrHcqM&t=3637s&ab_channel=EvansBohl (visited on 06/17/2024).

[84] *Post Process Materials - Unreal Engine 4.27 Documentation.* Epic Games. URL: https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/PostProcessEffects/PostProcessMaterials/ (visited on 06/23/2024).

[85] UnrealCG. *Toon Shader and Custom Shadow - UE4 Postprocess Tutorial [UE5 Valid].* June 25, 2020. URL: https://www.youtube.com/watch?v=mx_AvrZK3TA&t=207s (visited on 06/17/2024).