UNMASK: A THIRD-PERSON ROLE-PLAYING GAME AIMS TO RAISE SOCIAL AWARENESS FOR MENTAL HEALTH CONDITION ALEXITHYMIA

A Capstone Project Presented to the Faculty of the

School of Computer Studies

University of San Jose-Recoletos

Cebu City, Philippines

In Partial Fulfillment

Of the Requirements for the Degrees of

Bachelor of Science in Information Technology

by

Charles William D. Catingub

Joash Miguel G. Dandan

Niño Anthony L. Soldevilla

Engr. Vicente III F. Patalita

ADVISER

ACKNOWLEDGEMENT

The developers would like to express their gratitude to the valuable people that contributed through their continuous help and encouragement to allow the project to reach completion.

First of all, to the Almighty God for blessing them with an ample amount of time, diligence, and determination; to not only complete the project but to keep striving for glory.

To their adviser, Mr. Vicente Patalita III, for not only sharing his knowledge and experience but also his conviction and patience despite the difficulties they encountered all throughout the study.

To the panel, Mr. Roderick Bandalan, Mr. Roy Salares, Mr. Jeoffrey Gudio, and Mr. Leeroy Gadiane, for providing helpful observations in creating the project. And also for their honest feedback and critiques that help identify and fix errors or improve the project to them.

To family and friends, for simply being there, listening without judgment, and understanding them. Putting their whole faith and trust, the source of strength, motivation, and limitless love that drive the developers to excel more than before.

Lastly, to the developers, for giving it their all and having the tenacity not to back down when all seems lost and hopeless.

ABSTRACT

Unmask is a third-person game set on a story of a young boy with difficulty identifying and expressing emotions. The character wears an emotionless mask formed in a world of fantasy. It is a story-driven game that follows a traditional adventure quest as the player explores other places with interesting characters of different races, interacts with objects, fights creatures, and picks up items. This study aims to spread social awareness of a specific mental health condition by simulating a game that relates the symptoms of a person with this condition. This research covers developing a game that is fun to play and simultaneously delivers social awareness and knowledge. For the game's code to function, the developers used Visual Studio Code applying C# programming language to develop the game. 3D models and environment assets are created manually in Blender and Unity 3D. Developers used Clip Studio Paint, Adobe Photoshop, and Adobe Premiere for the game's icons, textures, and animated videos or infographics.

The sub-clinical failure to recognize and explain one's emotions and problems with emotional awareness, social attachment and interpersonal relationships are characteristics of the personality form known as Alexithymia. Only a few of the general population are diagnosed with it and well-informed of this complex mental health condition; this being in a spectrum could vary symptoms and how it affects the people

diagnosed with it. Being aware of this, at least, would raise underlying concerns, for understanding and expressing one's emotions are crucial parts of human experiences that commonly lead to misunderstandings or conflicts. *Unmask* helps address this unfamiliarity problem for players can experience and grasp the condition through immersion as the players progress in the game.

Keywords: Alexithymia, mental health, awareness, semi-immersive experience, story-driven game

ENDORSEMENT

PASSED

Table of Contents

LIST OF TABLES AND FIGURES

List of Tables

List of Figures

Figure

Figure

Figure

Figure

Figure

Figure

Figure

Figure

CHAPTER I

INTRODUCTION

Rationale

Video games are well known, particularly those geared towards children that can help them develop fundamental abilities like speed, agility, and game narrative.[1] They are unique tools for educational purposes like spreading social awareness, especially for players who barely understand various mental conditions. Adolescents may utilize the virtual world of video games as an opportunity to experiment with and subsequently regulate their emotions if they are unable to manage and express their feelings or if they are just unable to detect them (Alexithymia).[2]

Games can also offer a way for players to learn about their strengths and limitations as they advance through the levels of a game. Players can better understand themselves by observing their decisions and actions in these fictitious circumstances.[3]

The development of the character in the virtual environment might assist in mediating the troubling emotional experiences of adolescents, so they have meaning and are appropriate. Video games function as a medium for projecting and experiencing one's emotional existence via staging the emotional self, which accounts for adolescents' interest in video games.[4]

There is a requirement for a medical professional's diagnosis of the uncommon or unheard health condition. Despite this, providing information and learning about it does

not require any diagnosis. The game is, of course, backed with factual reports, data, and studies necessary to incorporate it with its validity.

Various treatments for Alexithymia mainly involve talk therapy and counseling. Other options include skills-based therapy, creative arts engagement, and reading emotional books or stories (Weiss, 2022).[5] With these options, *Unmask* ticks a checkmark to the three options stated. The game requires certain skill levels to progress while enjoying the creative and beautiful sceneries and assets and learning about its stories involving emotions within each stage. With these, recognizing the health condition should be evident as the player completes the game.

Significance of the Study

As video games and their community rise. This research aims to regulate social awareness about understanding mental health issues, specifically, Alexithymia and aims to remind and guide the players about understanding the validity of their emotions. With this context, video games can bring out feelings through their visual aspect, storylines, and audial ambiance. To reinforce the hypothesis, the group would develop a game that visualizes emotions in its gameplay and the progress of its story. From the 2D to 3D assets of the characters, map environment, items, and other interfaces. With fun and challenging gameplay for an overall immersive experience.

Obtaining feedback from the testers would have an expected result connection between video games and human emotions. It would also benefit the players, with the value of empathy usually overlooked these days. Completion shall relay that video games are stress-relievers, not stress-makers, and should be a fundamental concept for game developers in developing their projects.

Review of Related Works

The game *Florence* has a beautiful story that centers around a character that feels stuck, a feeling most of us have related to at one time or another. Florence's life changes when she meets a cello player named Krish as he helps her see the world through an entirely new lens and how she sees herself. [6]

Actual Sunlight is a game that has very guttural moments and, because of that, can often be challenging to play. The game's story is about love and depression, and the narrative does little to shy away from either of those topics. For those struggling with depression, this relatable experience can help them find a connection. It is an incredibly heartbreaking resolution, and the themes presented in Actual Sunlight are heavy and on the nose. [7]

That Dragon, Cancer, has won many awards for good reason. Based on a true story of how one family struggled with cancer, resounding loss, and the stages of grief. [8]

Hidden Folks has a unique hand-drawn art style that the players quickly get lost in the game. While not a narratively-focused game like some of the picks above, this is an excellent game to help decompress and hone down anxiety. [9]

Night in the Woods represents mental health. It tackles psychological disorders like bipolar, soothes depression, and allows players to learn new things about their current state. [10]

Project Objectives

The game aims to develop a 3D third-person RPG using the Unity Game Engine. The game will allow players to experience stages of explorations, battles, and emotions.

With that in mind, specifically, the developers aim to:

1. Create a linear story consisting of 4 chapters focusing on the player's experience as it progresses every stage.
2. Create 3D assets and an environment that represents different characters and stories.
3. Create an interactable environment for the player to explore.

4. Create User Interfaces according to their subject matter and their functionality as a mechanic.

5. Create a Pick-Up Mechanic, simple combat system, inventory system, health, shield, and stamina bar.
6. Integrate AI for characters (Enemy).
7. Integrate dialogue into the narrative and cutscenes.
8. Use Different tools (Blender, Unity3D, Clip Studio Paint, Adobe Photoshop, Adobe Premiere) to create assets for the game.
9. Develop the features and mechanics of the game using the Unity game engine.

Scope and Limitation of the Study

Unmask aims to develop a story-driven role-playing game for desktop devices that lets players experience different emotional stages. The player can explore a few areas of the environment and use basic movements such as walking, jumping, and running. The player has to stick with the story and take the correct course. The appropriate path and the story would be shown to the gamer. Players can interact with or pick up items that are on the chest. The object that the player picked up can be used or kept in the inventory. The player must achieve particular goals by following the narrative to the next stage. Every level begins in a specific location and only ends when the level's last task is successfully completed.

CHAPTER II

GAME DESIGN DOCUMENT

GAME NAME

UNMASK: A THIRD-PERSON ROLE-PLAYING GAME AIMS TO RAISE SOCIAL AWARENESS FOR MENTAL HEALTH CONDITION ALEXITHYMIA

GAME OVERVIEW

Game Concept

Unmask is a story-driven third-person perspective game set in a fantasy world. The game design aims to create 3D Low-poly assets, capturing the feel and atmosphere for each chapter's emotions that represent the story. A Korean drama series influenced the game, "It's Okay Not To Be Okay," conveying how everyone's feelings and emotions are valid. It is also to raise awareness about the genetic condition of Alexithymia. People diagnosed with this condition have difficulty identifying emotions and a seeming lack of feelings or an inability to express them. The game story teaches the players to be empathetic to the challenges and situations of the characters.

Genre

Unmask is an action-adventure game with a little element of platforming and a semi-open world. It employs a linear story with interactable objects, different environments, and an inventory system.

Target Audience

The target player for the game Unmask has the following characteristics:

1. Male or Female

2. Ten years old and above

3. Desktop Computer Gamer

Game Flow Summary

The game starts in the Main menu, where the player can choose between New Game, Continue, or Quit, as shown in the image below.

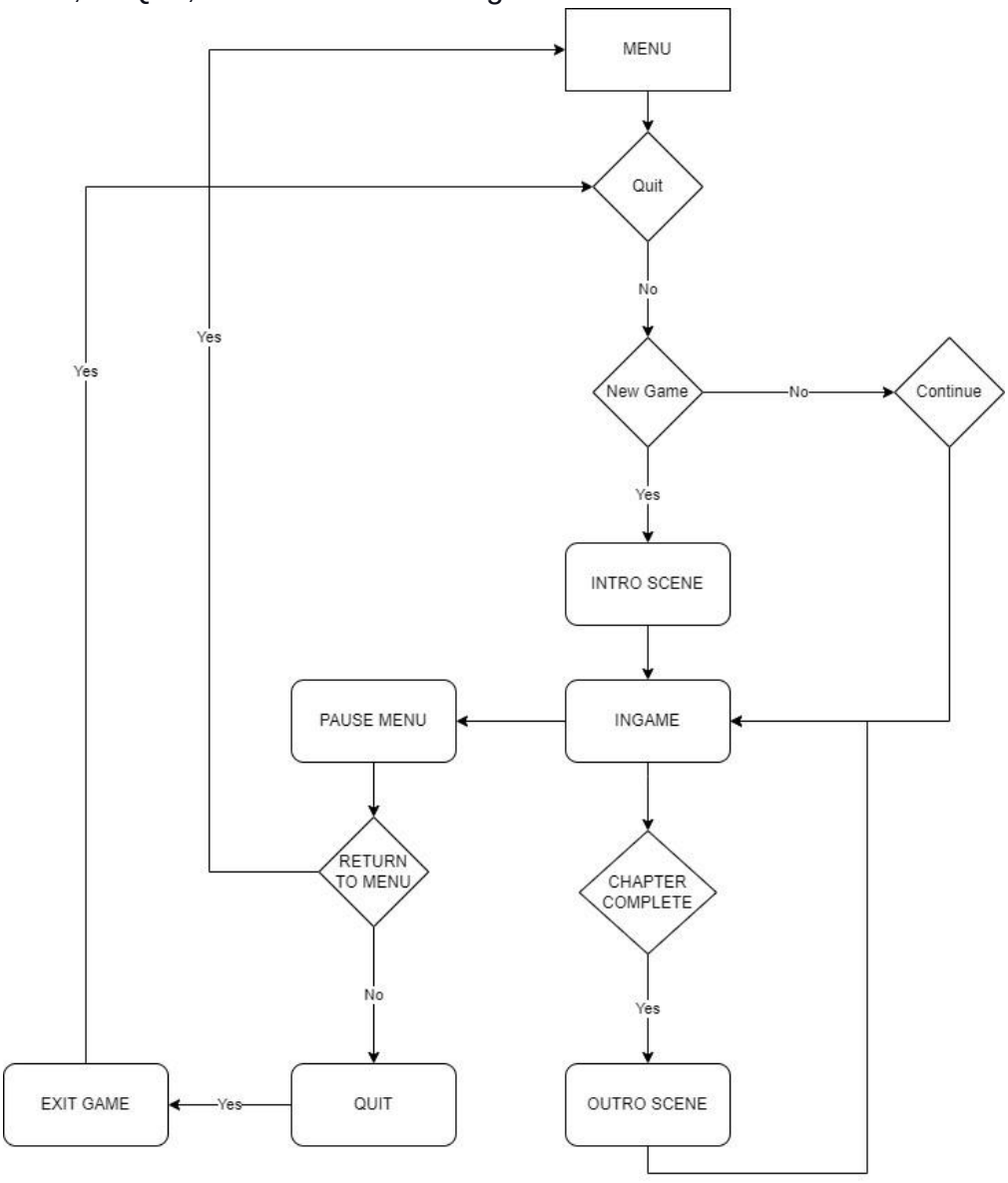

Figure 1: Game Flow Summary

FUNCTIONAL SPECIFICATIONS

GAME MECHANICS

Core Gameplay

The game's core gameplay consists of walking, shooting using the bow and arrow, dodging, attacking using a sword, collecting items for progression, and opening chests for loot. The game consists of four chapters with different tasks and challenges for the story to progress. The player must follow the quest to advance in the story, and each chapter has one boss to defeat.

Quests

Various quests push forward the progression of the level. There are five types of quests in the game.

- Destination Quest - A quest that requires the player to leave a certain level area. A wisp mainly accompanies it.
- Collect Quest - A quest that requires the pick-up of a list of items demanded by DAS.
- Defeat Quest - The quest requires the player to defeat a boss.

Units

Units are the AIs that can be interacted with in the game. A unit's default setting has four behaviors performed when in contact with its targets.

- Patrol - A behavior that allows the unit to walk through different points in the game to guard a specific area with either a sequential or random pattern.
- Chase - A behavior that allows the unit to run towards its target when it reaches a certain radius and is within a specified angle towards the unit.
- Attack - A behavior that allows a unit to engage and attack its target when it reaches a specific range from the unit.
- Idle - The default behavior of a unit when no other behavior is available for it to perform.

There are two types of Enemy Units in the game; Melee attacker and Range Attacker.

- Melee attacker- AI attacks the player at a close range to the player.
- Range Attacker- AI attacks the player from a certain distance.

Boss Battles

Each boss has different ways to defeat their game. The following are the characteristics of each boss.

- DAS - Can summon three ranged flowers.

- REAF - Can shoot fireballs and venom and summon spiders.
- ENGAR - Berserker mode (Increases size and health) ● PYAPH - Players will have to play a rhythm game.

Game Flow

The game flows into four chapters, each with three different objectives.

Prologue: Animated Scene

Chapter 1

Title: Tears or Raindrops?

Objectives

1 - Beat a water slime (Tutorial).

2 - Gather or collect the (3) flowers and beat the remaining water slimes.

3 - Beat the final boss.

Chapter 2

Title: I See Many Eyes Watching?

Objectives

1 - Slay the spiders.

2 - Run through the maze and find the exit.

3 - Beat the final boss.

Chapter 3

Title: Trust Nobody

Objectives

1 - Meet the bandits. (Memorize specific patterns) 2

- Beat the bandits.

3 - Beat the final boss.

Chapter 4

Title: Smile for Me

Objectives

1 - Meet Pyaph.

2 - Follow the beat and rhythm.

3 - Beat the final boss.

Epilogue: Animated Scene

MECHANICS

Movement

The player can move in a third-person perspective character.

Mouse

- Camera Movement – Mouse Drag: the movement is used for maneuvering the view from the player's perspective.
- Attack – Left Mouse Button: if a weapon is equipped, it triggers an animation for both bow and sword attacks.

Keyboard

- Move Forward – "W" key moves the player forward.
- Move Left – "A" key allows the player to strafe on the left side.
- Move Backward – "S" key moves the player backward.
- Move Right – "D" key allows the player to strafe on the right side.
- Jump – "Space" key propels the player upwards for a jump.
- Dodge – "F" key allows the player to distance himself from an incoming attack.
- Sprint - "LeftShift" key boosts the player's movement but can cost stamina.

Weapons

Players can use two different weapons, swords and bows, to defeat enemies from a particular range.

Interactables

The player can discover interactable objects, such as levers, to open hidden parts of the maps. These remote parts of the maps may contain potions to aid the player in his future battles.

Items

The player can pick up items from the world, and a prompt message will pop on the screen by pressing the "E" key. Items and quest items can be seen in the player inventory by pressing the "I" key.

Inventory

The inventory stores the items picked up by the player. It allows the player to view the picked-up items and quest items.

Stats Bar

The player will have three different bars with different functionalities. One is for health, a red bar, Gray for the armor, and yellow for stamina. This will help the player track their life which makes a crucial element of the game.

As for all types of enemies, they will only have a single status bar, but other special characters will have a much larger life than the lesser units to give the player a

challenge.

Combat

The combat varies depending on the player's weapon; the player can choose two types of combat experience depending on the enemy's range of combat. Players can use swords for melee attacks and switch to a bow and arrow to get an advantage of the range. Though the player can use a bow, its arrow is not infinite. Players must conserve their arrows for important battles such as boss battles.

For the enemies, it has the same setting for Melee and Ranged attacks. However, they do not require pick-up items and only have their default attacks. In addition, only enemy Bosses can change their attack patterns with their different states as their Health drops.

Units also have a hit mechanic that, when receiving a certain amount of damage, will play a hit animation and be knocked out for a specified number of seconds.

The enemies can also detect certain characters depending on their radius and distance from the target. It could vary from a short-sighted unit that can only detect enemies in front or a hyper-sensitive unit that can detect enemies from 360 degrees.

Gameplay Mechanics

- The player is able to move at a level's start point.
- Any progress from a previous level will be loaded into the current level.
- Quests and dialogue move the narrative and the player toward the final boss.
- Items can be picked up during the duration of each level.

● Enemies guard certain areas to be slain or evaded depending on the quest.

● After the boss is defeated, the level ends by learning its backstory.

● The game can be saved by checkpoints only.

STORY, SETTING, AND CHARACTER

Story and Narrative

The story starts with a young boy born with difficulty understanding emotions. He lived in a world where people wore magical masks that could shift to different emotions depending on the wearer's feelings. The boy struggles to understand why his mask is different from the other people in his village, a mask that does not show any emotion. He was cast out from society and wandered alone, trying to survive beyond the woods full of beasts and monsters. He learned to adapt and survived countless near-death situations.

Then one day, he encounters a wisp while venturing into a swamp area. There he was told about why people of different races always wear masks. Long ago, an old evil fiend whose magic stole faces, the races began to use wooden masks from a magical tree to protect them from the evil fiend and keep them at bay. Little did they know, the old evil magic was already gone long ago, yet the tradition of wearing the mask was still a big part of their daily lives. Some individuals accidentally overused the mask, and some took advantage of its magic and used it to fuel their emotions as they secluded themselves from others. In the end, the wisp had no idea why the boy wore an emotionless mask. He

suggests accompanying the boy to venture and look for these specific people with a strong sense of individual emotions.

Game World

Unmask has elements of a fantasy world broken down into four different locations.

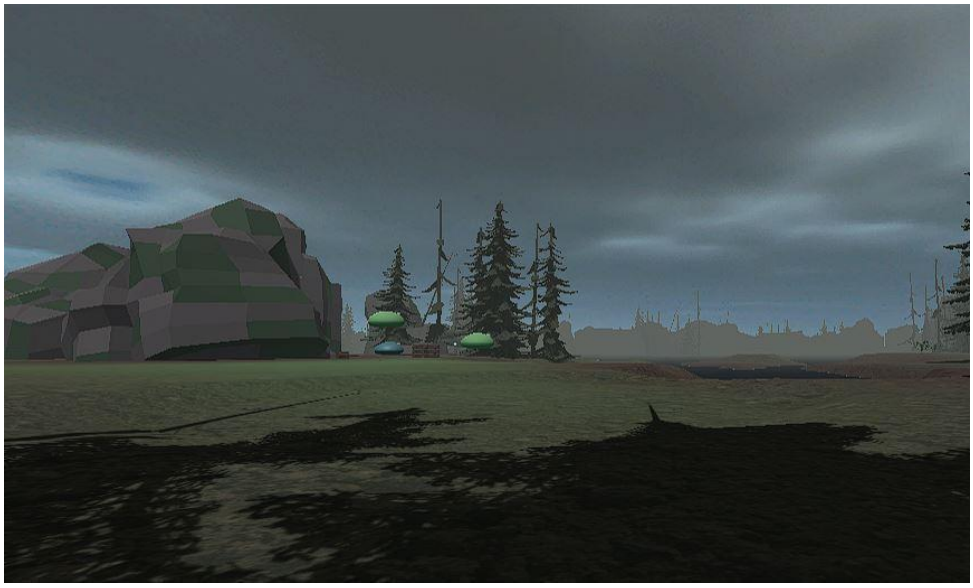

Figure 2: Chapter 1 Swamp Area

As shown in the image above, the first chapter has a swampy area and foggy gloomy weather. It consists of hidden areas of moving boulders and run-down small cabins.

Figure 3: Chapter 2 Maze Forest

The image above shows that the second chapter leads to a maze forest. The area is filled with Tall trees and crimson fog limiting the player's FOV making it much more difficult to find the exit.

Figure 4: Chapter 3 Bandit's Tavern

The image above shows the first small area for the third chapter. The area is filled with bandits guarding the abandoned tavern and with a trapdoor that is the hidden entrance to the cave.

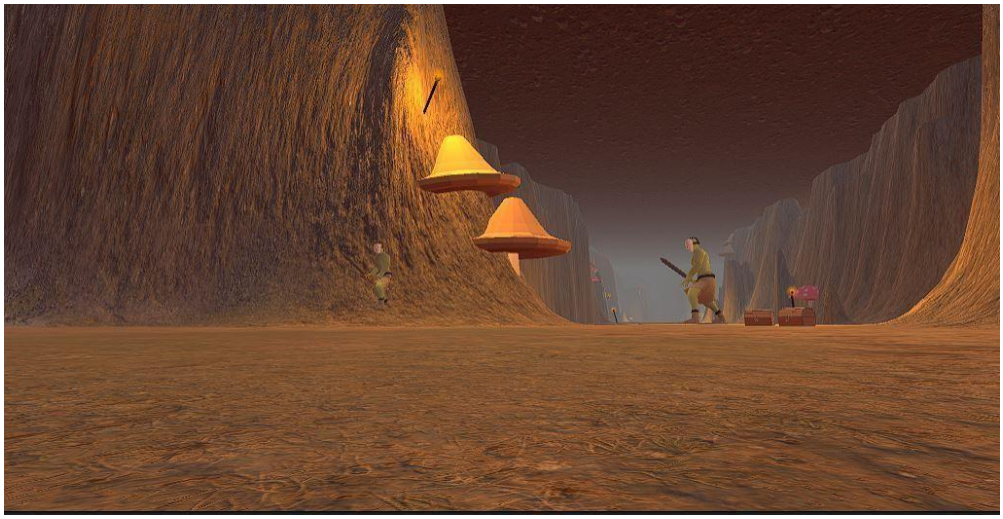

Figure 5: Chapter 3 Bandit's Cave

As shown in the image above, the second map for chapter 3 is a cave system. Mushrooms are common inside the cave, and pits as well, and contain constantly moving platforms.

Figure 6: Chapter 4 Village

As shown in the image above, is the village map for chapter 4. The village is full of people and festivals. Consists of small houses, small farming fields, and festival props.

Characters

Young Boy

He is the main character of the game. A young boy is forced out of his village to find people who can help him understand emotions.

The Wisp

Wisp is the magical friend helping the primary character search for the individuals that might help him understand what emotions are.

Das

Das is a dwarf and is often bullied by his friends for being an orphan. He was the last orphan not even adopted. He never experienced the warmth of love from friends and family. It deals with bullying, loneliness, and abandonment.

Reaf

Reaf is a high elf and is insensitive in her words and actions and is often neglected by her strict parents, seeking attention and wanting validation. As she grew old, she was labeled and ridiculed and was not given a chance to change. She isolated herself from the public

and lost her self-confidence. Deals with insensitive words or actions, self-confidence, and gatekeeping.

Engar

Engar suffered poverty and was malnourished at a young age. He grew up in a slump area and survived the so-called "survival of the fittest." Oathed himself to become the wealthiest man in town and formed a group of bandits, and became the leader. He would meet Pyaph, who would then manipulate the group to its demise through the betrayal of Engar's subordinates. Loathe Pyaph and seeks his vengeance. It deals with societal issues about the needy, value of trust, weighs, and the consequences of seeking revenge.

Pyaph

Pyaph easily entertained people as a clown or performer. He was an eloquent speaker and storyteller and wanted to become the best clown in town and deliver consistently outstanding performances. People are not amused anymore by him and his performances. He Became desperate and went mad due to his hunger. It deals with knowing one's true self and doing things loved by the self and not for others.

VISUAL SYSTEM

The User Interface design created in black & white color design is a minimal approach.

Figure 7: Menu UI

As seen in the image above, the menu consists of Start, Continue and Quit buttons. Start button creates a new game starting from the 1st chapter. Continue button loads the last checkpoint the player activated. Quit button, closes the game completely.

Figure 8: Inventory Bag Icon

The above image shows the inventory icon that can be viewed on the lower right side of the screen.

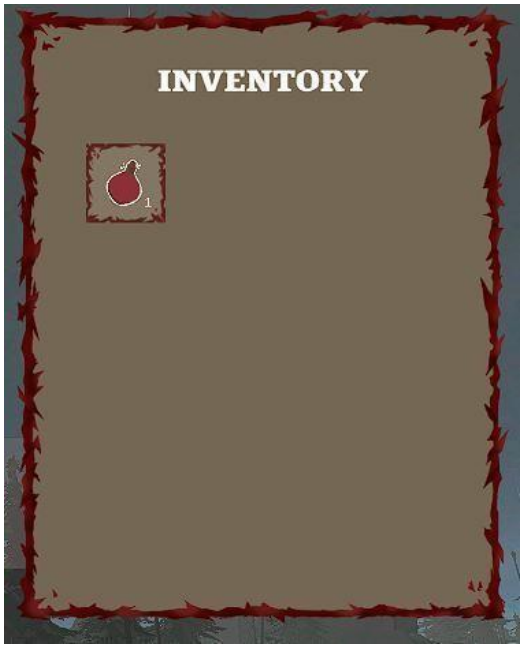

Figure 9: Inventory Panel

Upon triggering the key bind for the inventory, it can be toggled to view the items picked up by the player, as shown in the image above.

Figure 10: Potion Icon

The image above shows the potion icon that can be viewed on the middle left side of the screen.

Figure 11: Arrows Icon

The image above shows the arrows icon that can be viewed on the middle left side of the screen.

Figure 12: Quest UI

The image above shows the Quest UI which can be found in the left-center corner of the screen.

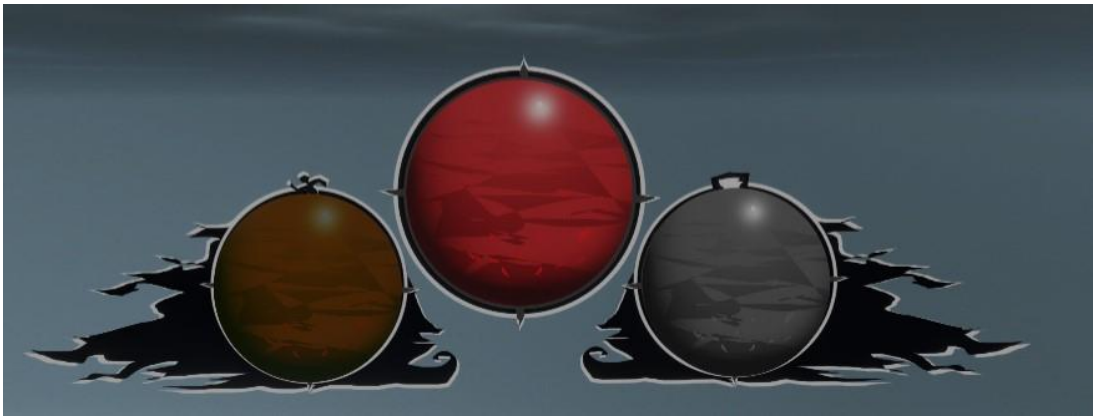

Figure 13: Status Bars

The image above shows the player status bars which can be found in the lower-left corner of the screen. The red bar represents Health, the yellow is for stamina and the gray is for the shield.

Figure 14: Marigold Icon

The image above shows the Marigold icon that can be viewed on the inventory panel once picked.

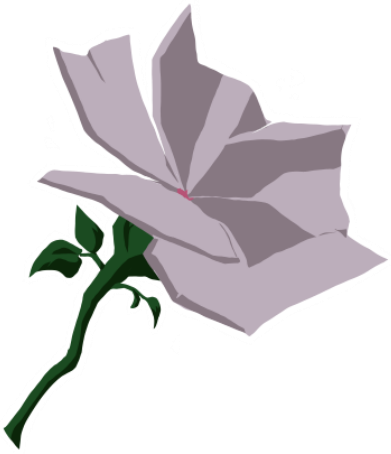

Figure 15: Rosemallow Icon

The image above shows the Rosemallow icon that can be viewed on the inventory panel once picked.

Figure 16: Cornflower Icon

The image above shows the Cornflower icon that can be viewed on the inventory panel once picked.

TECHNICAL

Target Hardware

Hardware: Personal Computer - The platform used in building and Unmask.

Processor: Intel i3 6th Gen or AMD equivalent - 6th Gen - The processors that can handle and perform multiple operations simultaneously.

Operating System: Windows 10 - The chosen operating system for Unmask.

Ram: 4GB+ - The minimum RAM or Random-Access Memory to play the game.

Graphics: AMD and Nvidia dedicated Graphics cards - The dedicated graphics cards can handle loads of graphics.

Controllers: Mouse and Keyboard - The controllers used for interacting and navigating the game.

Storage: 5 GB free - The estimated storage is to be free when downloading the game.

Software Development Tools

Unity3D Game Engine - A software developed by Unity Technologies as a gaming engine for building games.

Visual Studio Code 2019 - A source code editor used in programming the back-end and front-end of games and applications.

Blender - An open-source 3D software used in creating and texturing 3D models.

Adobe Photoshop - A software developed by Adobe for editing and digitally illustrating images.

Adobe Premiere - A software developed by Adobe for animating videos and infographics.

Architectural Design

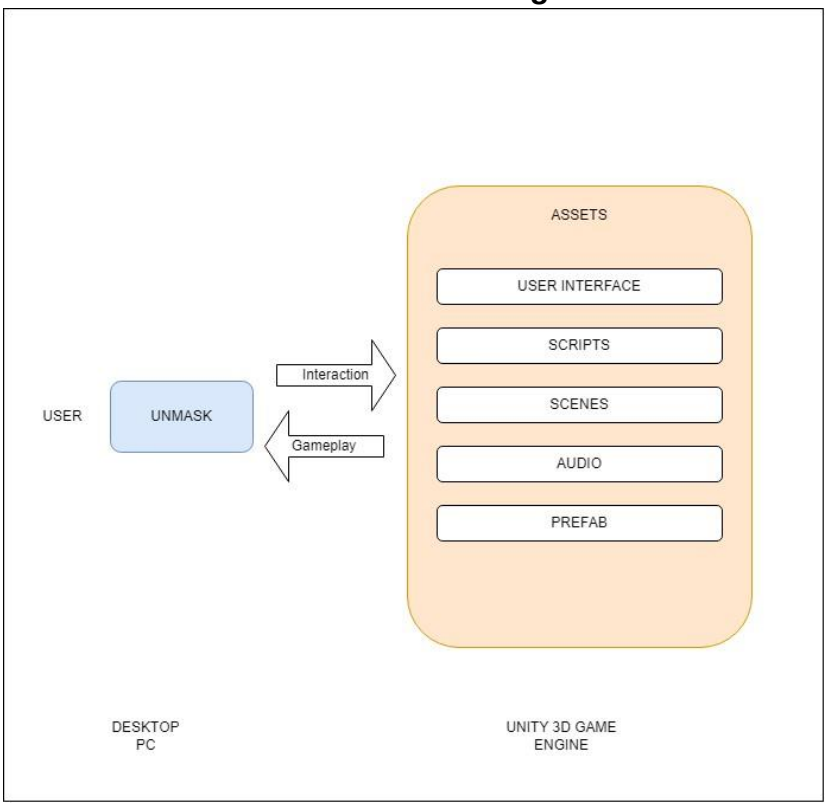

Figure 17: Unmask Architectural Design

An immersive and playable game requires excellent and steady development.

The game's flow and functionality are supported by an easy-to-follow script structure. Visual Studio Code uses C# language and is used to program the scripts in the game. The various assets are created using Blender for the 3D models and rigging, Adobe Photoshop for textures, Premiere for animated scenes, and Unity 3D to provide convenience for animation and the final touch-up for the design.

Game Art

Images below are the concept art for the level design, User Interface, and Player view.

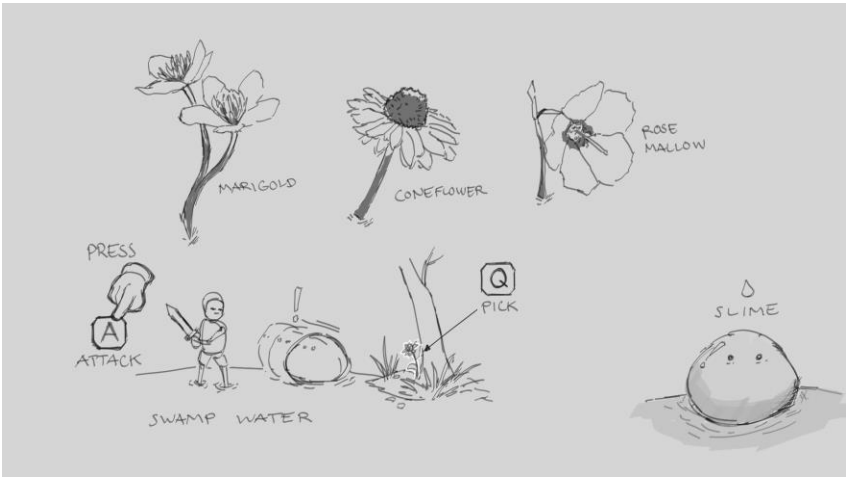

Figure 18: Unmask Chapter 1 Concept Design

The image above is the first chapter’s concept art. The original concept sketch for the chapter 1 characters and interactables.

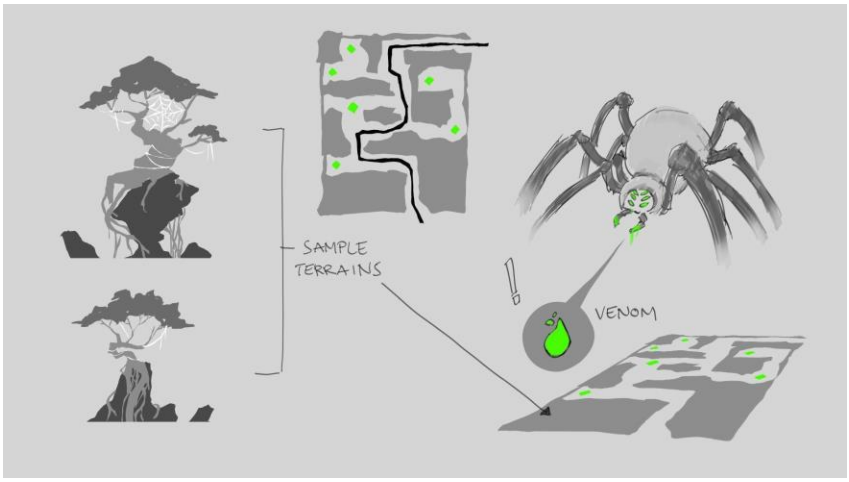

Figure 19: Unmask Chapter 2 Concept Design

The image above is the original concept art for the chapter 2 environment. The original maze divider was later changed to a forest maze.

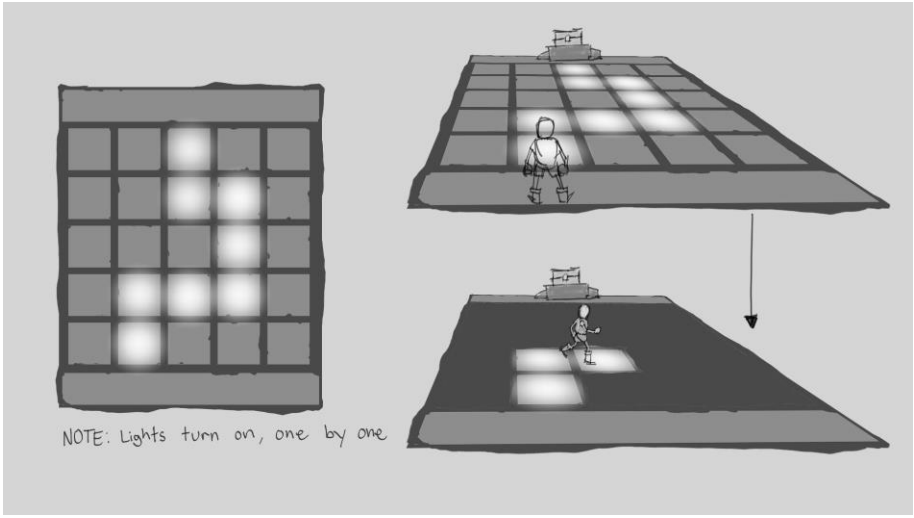

Figure 20: Unmask Chapter 3 Concept Design

The image above is the original concept art for the platform game for the chapter 3 environment.

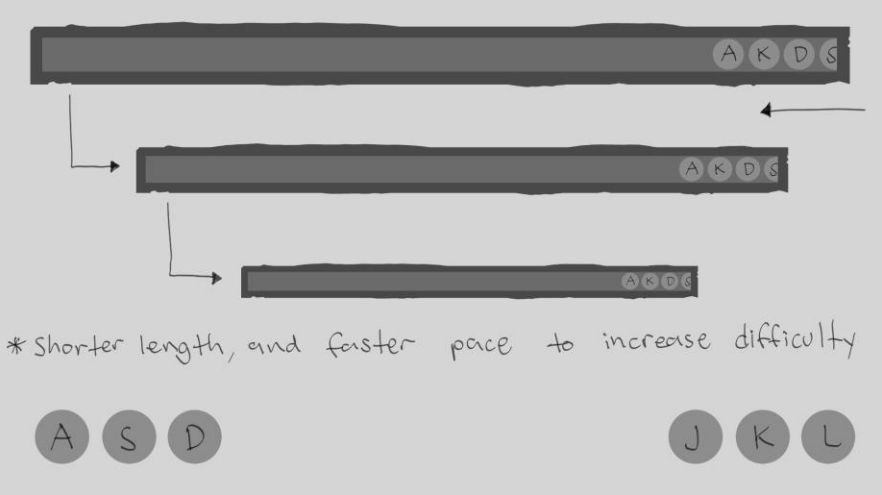

Figure 21: Unmask Chapter 4 Concept Design

The image above is the original and early concept art for the rhythm game for the chapter 4 final battle.

Figure 22: Unmask All Character Concept Design

The image above is the first concept design for the characters in the game. The original concept sketch for WISP, BOY, DAS, ENGAR, PYAPH, and REAF.

Figure 23: Unmask Chapter 1 Environment Concept

The image above is the first concept design for the first chapter. A rainy and foggy swamp area.

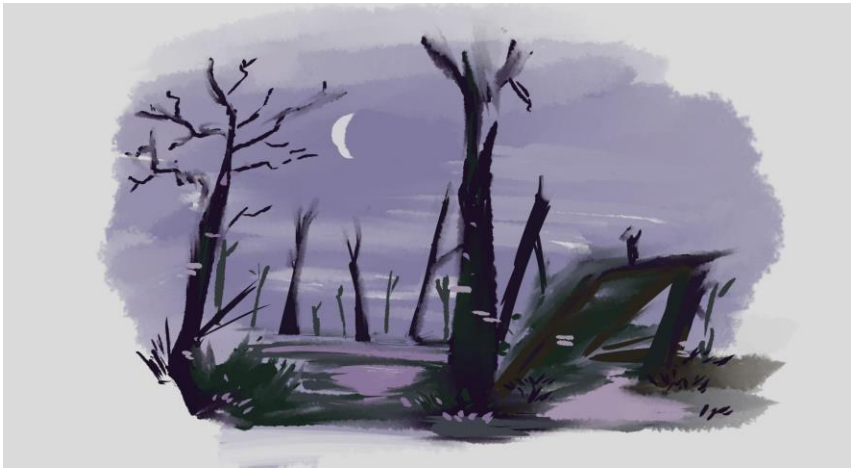

Figure 24: Unmask Chapter 2 Environment Concept

The image above is the first concept design for the second chapter. A dark forest maze.

Figure 25: Unmask Chapter 3 Environment Concept

The image above is the first concept design for the third chapter. Hill and dry land area.

Figure 26: Unmask Chapter 4 Environment Concept

The image above is the first concept design for the fourth and last chapter. A bright and sunny area.

CHAPTER III

SOFTWARE DEVELOPMENT AND TESTING

The project's general teamwork in creating a role-playing game stemmed from the developers' tenacity and perseverance. Although enthusiasm alone only works with the correct framework, it is only possible to fully describe what makes a game, or an existing project, work. These are types of equipment that were used throughout the game's development. The many tools and platforms are covered in this chapter. Implemented and incorporated into the various game elements will be displayed and demonstrated, sparkling light on the game's development.

DEVELOPMENT SOFTWARE PLATFORMS, DEVELOPMENT ENVIRONMENTS, AND TOOLS

The *Unmask* game was developed with Windows 10 OS. Unity 2020.3.11f1 is the version used for creating the game. The intended platform for the game is PC and, therefore, will be the only compatible platform. C# is the programming language used, and Visual Studio 2019 is the programming environment used to implement said language. Other software used in the development of *Unmask* is as follows:

Blender

Blender is a 3D computer graphics program that is open-source and free. It is used to make animated movies, visual effects, artwork, 3D-printed models, motion graphics, 42 interactive 3D apps, virtual reality, and video games (Blender Foundation, 2022).[11] In addition, the developers used Blender to create and design 3D characters and environment assets.

Clip Studio Paint

Clip Studio Paint is a graphics software used for digital illustrations (Clip Studio Paint, 2022).[12] The developers used Clip Studio Paint to do the lineart, colors, and shading for the 2D graphics.

Adobe Photoshop

Adobe Photoshop is a graphics, painting, and 3D artwork program compatible with Windows and macOS (Official Adobe Photoshop | Photo and Design Software, 2022).[13] The developers used Adobe Photoshop to design the graphics of the user interface and the game's logo.

Adobe Premiere Pro

Adobe Premiere Pro is a video editing software developed by Adobe Inc. (Professional Video Editing Software | Adobe Premiere Pro, 2022).[14] The developers utilized this to edit and animate videos or infographics of the game.

Unity Game Engine

Unity Game Engine is a software that powers the game engine Unity Technologies. Unity is a well-liked option for game development since it offers comprehensive support for a 43 wide range of game kinds, including 2D and 3D games (Technologies, 2022).[15] C# is the preferred programming language for Unity and may be used with various types of software. C# was chosen as the programming language for Unity due to its adaptability when employed with many sorts of software. However, Unity is the chosen platform used to create the game by developers because of its sophisticated tools and features.

Visual Studio 2019

Visual Studio 2019 is one of Microsoft's integrated development environments. (IDE) Used for developing computer programs (Microsoft, 2022).[16] The developers used Visual Studio 2019 to program the game's front-end and back-end.

DEVELOPMENT PROCESS AND USABILITY TESTING

Pre-Production

The game's goal runs through the story of a young boy on his adventure to identify feelings and emotions. He will encounter different four basic types of emotions through

different races at each level. Every stage has its own unique level design based on the boss' emotion on the level.

Game Concept

Unmask is a story-driven third-person perspective game set in a medieval fantasy world. The game design aims to create 3D Low-poly assets, capturing the feel and atmosphere for each chapter's emotion that represents the story. It consists of four chapters, each with its unique character and story. The player will have the chance to meet these unique characters, unfolding their backstories as it progresses in their respective environments and levels.

Character Design

In designing the character, the developers created characters based on the medieval era with a mix of fantasy. All characters have different visuals, from height to skin color, yet they all have one thing in common. They all wear masks, and each character has a unique mask representing their strongest emotions.

Figure 27: Unmask Main Character Model

After conceptualizing the character, the 3D model is created using Blender, as shown in the image above. Default materials are added to the specific parts used for texturing. The character designs are more minimalist to fit the character's personality.

Figure 28: Unmask Main Character Model Rig

The rigging also comes next from Blender, but other rigs are created in Mixamo along with its animation, as shown in the above image. Some models are only rigged, and the animation is done in Unity.

Figure 29: Spider 3D

Other non-humanoid creatures in the game require a different bone rigging. Spiders and slime have different bones and are manually rigged for them to animate.

Structure Design

In designing the structures, the developers wanted to make the environment related to the boss' emotions in the chapter. For the game's terrain, we used Unity 3D terrain to make the whole environment. Custom textures are also used to paint on the terrain to add details.

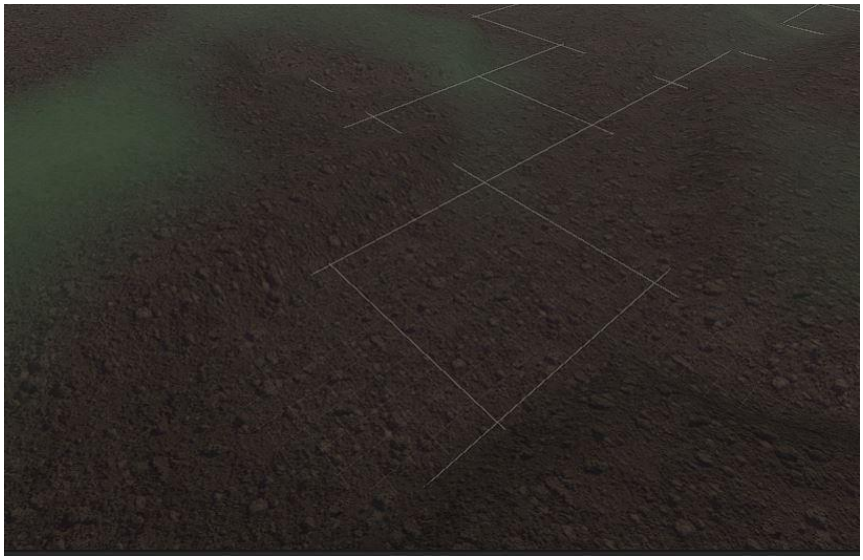

Figure 30: Chapter 1 Terrain Texture

The image above shows the terrain and different texture. The developer chose these textures to make it not realistic to keep the game style and design similar.

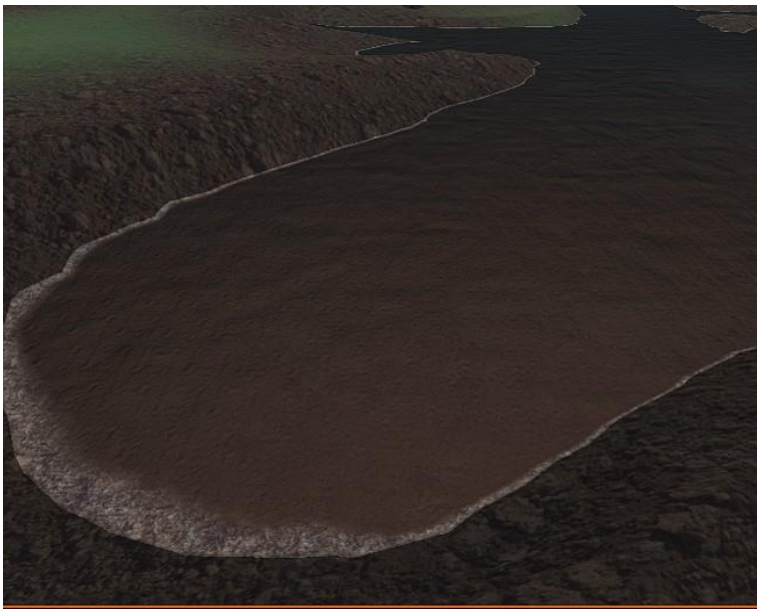

Figure 31: Chapter 1 Terrain Water Texture

The image above shows the water texture of the swamp environment adding detail to the environment.

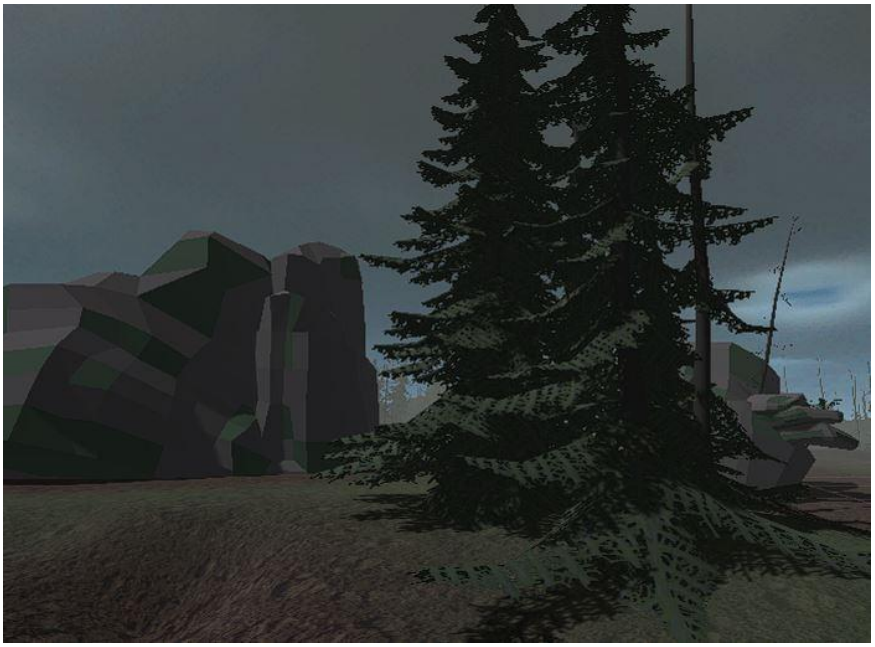

Figure 32: Chapter 1 Environment Object

The image above shows the rocks and trees for the swamp environment. Adding objects like trees and rocks gives more environmental detail.

Design

The icons used in the game consist of created and edited images. The created images are used when an icon gets assembled in Unity. Each UI design layout is hand drawn with a distinct art style.

Figure 33: Chapter 1 Marigold Art

The image above shows the Marigold artwork for the icon in chapter 1. The developers maintain a specific art style for the icons and UI.

Production

Player Script

```
//Move character by WASD
input.x = Input.GetAxis("Horizontal");
input.y = Input.GetAxis("Vertical");

animator.SetFloat("InputX", input.x);
animator.SetFloat("InputY", input.y);

///ACTIONS////
if (Input.GetKeyDown(KeyCode.Space))
{
    animator.SetTrigger("Jump");
}

//Sprint action and depletes stamina
if (Input.GetKeyDown(KeyCode.LeftShift) && playerStamina.CurrentStamina >= sprintStamina)
{
    isSprinting = true;
    animator.SetBool("Sprint", true);
}

if (isSprinting && playerStamina.CurrentStamina >= sprintStamina)
{
    playerStamina.CurrentStamina -= sprintStamina * Time.deltaTime;
    jogSound.Play();
}
else if (playerStamina.CurrentStamina < sprintStamina)
{
    isSprinting = false;
    animator.SetBool("Sprint", false);
    jogSound.Stop();
}
```

Figure 34: Player Locomotion Script

The code snippet above shows the basic movement of the player. The movement variables are input.x and input.y for getting horizontal and vertical axis. Sprinting action and jumping depletes stamina.

Combat Script

```
////COMBAT////
if (Input.GetKeyDown(KeyCode.Mouse0) && playerStamina.CurrentStamina >= attackStamina && !isAttacking)
{
    if (swordEquip.sword_is_equipped == true)
    {
        isAttacking = true;
        animator.SetTrigger("Attack");
        attackSound.Play();
        playerStamina.CurrentStamina -= attackStamina;
    }
}
```

Figure 35: Combat Script

The code snippet above shows the combat action for the player. Players can only perform attacks only when weapons are currently equipped. This also applies to player bow script.

Player Projectile Script

```
//Players arrow projectile
Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag == "Enemy")
    {
        other.GetComponent<EnemyAI>().EnemyTakeDamage(damage);
        EnemyCurrentHealth = other.gameObject.GetComponent<EnemyAI>().CurrentHealth;
        EnemyMaxHealth = other.gameObject.GetComponent<EnemyAI>().MaxHealth;
        EnemyName = other.gameObject.GetComponent<EnemyAI>().name;
        Destroy(gameObject);
    }
}
```

Figure 36: Player Arrow Projectile

The code snippet above shows the Player projectile/s script. If the projectile hits a game object “Enemy” It sends damage to the enemy health script and destroys the game object in the process.

Player Weapon Switching Scripts

```
///WEAPON SWTICHING///
if (Input.GetKeyDown(KeyCode.Alpha1))
{
    if (!swordEquip.sword_is_equipped && !bowEquip.bow_is_equipped)
    {
        sheathSound.Play();
        animator.SetTrigger("EquipSword");
        swordEquip.sword_is_equipped = true;
    }
    else if(swordEquip.sword_is_equipped == true)
    {
        animator.SetTrigger("UnequipSword");
        swordEquip.sword_is_equipped = false;
    }
}

if (Input.GetKeyDown(KeyCode.Alpha2))
{
    if (!bowEquip.bow_is_equipped && !swordEquip.sword_is_equipped)
    {
        animator.SetTrigger("EquipBow");
        bowEquip.bow_is_equipped = true;
        aimCamera.gameObject.SetActive(true);
        crossHair.SetActive(true);
    }
    else if(bowEquip.bow_is_equipped == true)
    {
        animator.SetTrigger("UnequipBow");
        bowEquip.bow_is_equipped = false;
        aimCamera.gameObject.SetActive(false);
        crossHair.SetActive(false);
    }
}
```

Figure 37: Player Weapon Switching Script

The code snippet above shows the switching condition for the weapons. Players can only equip one sword at a time. Players will have to unequip the currently equipped weapon to equip the other weapon.

```
//Equips sword to he players hand
for (int i = 0;i < sword.Count; i++) {
    if (sword_is_equipped)
    {
        sword[i].position = sword_eq.position;
        sword[i].rotation = sword_eq.rotation;

    }
    else
    {
        sword[i].position = sword_ueq.position;
        sword[i].rotation = sword_ueq.rotation;
    }
}
```

Figure 38: Player Sword Weapon Switching Script

The code snippet above shows the switching condition for the sword. Players can't equip a bow while the sword is equipped. This also applies to the player bow script.

```
//Change player's sword into steel
Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if (other.tag == "Player")
    {
        steelSword.gameObject.SetActive(true);
        ironSword.gameObject.SetActive(false);
        other.GetComponent<PlayerStats>().Damage = damage;
        Destroy(gameObject);
    }
}
```

Figure 39: Player Sword Weapon Type Steel Script

The code snippet above shows the switching condition for changing sword variants.

```
//Change player sword into iron sword
Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if (other.tag == "Player")
    {
        ironSword.gameObject.SetActive(true);
        woodenSword.gameObject.SetActive(false);
        other.GetComponent<PlayerStats>().Damage = damage;
        Destroy(gameObject);
    }
}
```

Figure 40: Player Sword Weapon Type Sword Script

The code snippet above shows the switching condition for changing sword variants.

Player Health Display Script

```
//Display player's health
Unity Message | 0 references
void Update()
{
    MaxHealth = player.MaxHealth;
    currentHealth = player.CurrentHealth;
    PlayerHealthBar.fillAmount = currentHealth / MaxHealth;
}
```

Figure 41: Player Health Display Script

The code snippet above shows the health of the player display script.

Player Inventory Scripts

```
2 references
public void AddItem(ItemObject _item, int _amount)
{
    for (int i = 0; i < Container.Count; i++)
    {
        if(Container[i].item == _item)
        {
            Container[i].AddAmount(_amount);
            return;
        }
    }
    Container.Add(new InventorySlot(database.GetId[_item] ,_item, _amount));
}
```

Figure 42: Player Inventory Add Item Script

The code snippet above shows adding item script.

```
2 references
public void Save()
{
    string saveData = JsonUtility.ToJson(this, true);
    BinaryFormatter bf = new BinaryFormatter();
    FileStream file = File.Create(string.Concat(Application.persistentDataPath, savePath));
    bf.Serialize(file, saveData);
    file.Close();
}
```

Figure 43: Player Inventory Save Script

The code snippet above shows how the items are saved when inside the inventory.

```
2 references
public void Load()
{
    if (File.Exists(string.Concat(Application.persistentDataPath, savePath)))
    {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(string.Concat(Application.persistentDataPath, savePath), FileMode.Open);
        JsonUtility.FromJsonOverwrite(bf.Deserialize(file).ToString(), this);
        file.Close();
    }
}
```

Figure 44: Player Inventory Load Script

The code snippet above shows how the items inside the inventory are loaded on the scene.

```
//Counts arrow from inventory
for (int i = 0; i < inventory.Container.Count; i++)
{
    if (inventory.Container[i].item.name == "Arrow")
        counter.text = inventory.Container[i].amount.ToString();

    if (inventory.Container[i].amount > 0 && inventory.Container[i].item.name == "Arrow")
    {
        var tempColor = image.color;
        tempColor.a = 1f;
        image.color = tempColor;
    }
    else if (inventory.Container[i].amount <= 0 && inventory.Container[i].item.name == "Arrow")
    {
        var tempColor = image.color;
        tempColor.a = .5f;
        image.color = tempColor;
    }
}
```

Figure 45: Player Inventory Counts Arrow Script The code snippet above shows how it counts arrows on the inventory.

Platform Script

```
public class attachplayer : MonoBehaviour
{
    public GameObject player;

    //attach player to the moving platform
    Unity Message | 0 references
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "Player")
        {
            player.transform.parent = transform;
        }
    }
    Unity Message | 0 references
    private void OnTriggerExit(Collider other)
    {
        if (other.gameObject.tag == "Player")
        {
            player.transform.parent = null;
        }
    }
}
```

Figure 46: Platform Script

The code snippet above shows the platform script. When the player stands on the moving platform the player becomes a child of the platform on trigger enter.

Camera Shaking Effect Script

```
public void shake()
{

    //shakes camera when notes hit collider
        start = false;
        StartCoroutine(Shaking());

}

1 reference
IEnumerator Shaking()
{
    Vector3 startPosition = transform.position;
    float elapsedTime = 0f;

    while(elapsedTime < duration)
    {
        elapsedTime += Time.deltaTime;
        float strength = curve.Evaluate(elapsedTime / duration);
        transform.position = startPosition + Random.insideUnitSphere * strength;
        yield return null;
    }
    transform.position = startPosition;
}
```

Figure 47: Camera Shake Script

The code snippet above shows the camera script. When the player misses a note, the screen will shake abruptly.

AI Behavior

Enemy AI Scripts

```
//Check for sight and attack range
playerSightRange = Physics.CheckSphere(transform.position, sightRange, isPlayer);
playerAttackRange = Physics.CheckSphere(transform.position, attackRange, isPlayer);

if (typeOfEnemy == enemyType.melee) {
    if (!playerSightRange && !playerAttackRange) Patrolling();
    if (playerSightRange && !playerAttackRange) ChasePlayer();
}
if (playerSightRange && playerAttackRange) AttackPlayer();
```

Figure 48: Enemy AI Script

The code snippet above shows the AI attack range script. The script uses physics.checksphere to detect the player.

```
private void Patrolling()
{

    if (!walkPointSet) SearchWalkPoint();


    if (walkPointSet)
    {
        enemy.SetDestination(walkPoint);
        animator.SetTrigger("Walk");
    }

    Vector3 distanceToWalkPoint = transform.position - walkPoint;

    //walkpoint reached
    if (distanceToWalkPoint.magnitude < 1f)
        walkPointSet = false;

}
```

Figure 49: Enemy AI Script

The code snippet above shows AI patrolling script.

```
private void SearchWalkPoint()
{
    //Calculate random point in range
    float randomZ = Random.Range(-walkPointRange, walkPointRange);
    float randomX = Random.Range(-walkPointRange, walkPointRange);

    walkPoint = new Vector3(transform.position.x + randomX, transform.position.y, transform.position.z + randomZ);

    if (Physics.Raycast(walkPoint, -transform.up, 2f, ground))
        walkPointSet = true;


}
```

Figure 50: Enemy AI Script

The code snippet above shows AI searching walk point.

```
private void ChasePlayer()
{
    if (!alreadyAttacked) {
        enemy.SetDestination(player.position);
        animator.SetTrigger("Walk");
    }


}
```

Figure 51: Enemy AI Script

The code snippet above shows AI chasing the player if detected.

```
public void DamagePlayer()
{
    float damageDistance = Vector3.Distance(player.position, this.transform.position);
    if (typeOfEnemy==enemyType.melee) {
        if (damageDistance <= attackRange + .5f)
        {
            playerStat.PlayerTakeDamage(Damage);
        }
    }


}
```

Figure 52: Enemy AI Script

The code snippet above shows AI will send damage to the player.

```
private void AttackPlayer()
{
    //make sure enemy doesnt move
    transform.LookAt(player.position);
    if (typeOfEnemy == enemyType.melee)
    {
        enemy.SetDestination(transform.position);
        //transform.LookAt(player.position);
    }

    if (typeOfEnemy == enemyType.range)
        transform.LookAt(player.position);


    if (!alreadyAttacked)
    {
        if (typeOfEnemy == enemyType.range)
        {
            shoot();
            alreadyAttacked = true;
        }else if (typeOfEnemy == enemyType.melee)
        {
            alreadyAttacked = true;
        }
        attack.Play();
        animator.SetTrigger("Attack");
        Invoke(nameof(resetAttack), timeBetweenAttacks);
    }
}
```

Figure 53: Enemy AI Script

The code snippet above shows AI will attack the player depending on the range of the enemy attached to it.

```
private void resetAttack()
{
    alreadyAttacked = false;
}

2 references
public void EnemyTakeDamage(float PlayerDamage)
{
    CurrentHealth -= PlayerDamage;
    if (CurrentHealth<=0)
    {
        enemy.isStopped = true;
        this.enabled = false;
        death.Play();
        animator.SetTrigger("Death");
        Invoke(nameof(DestroyEnemy), 4);
    }
    else
    {
        hit.Play();
        animator.SetTrigger("Hit");
    }
}
```

Figure 54: Enemy AI Script

The code snippet above shows AI will reset the attack and takes enemy damage from the player

```
1 reference
public void shoot()
{
    Instantiate(projectile, ProjectileSpawn.position, ProjectileSpawn.rotation);
}
```

Figure 55: Enemy AI Script

The code snippet above shows AI will shoot projectiles toward the player's position.

Animation

The animation for the units, including the player, follows a default set of behaviors and transitions. All the other units follow the same set, diverting special cases for boss battles or other purposes. The player has a default set but is not similar to the other units.

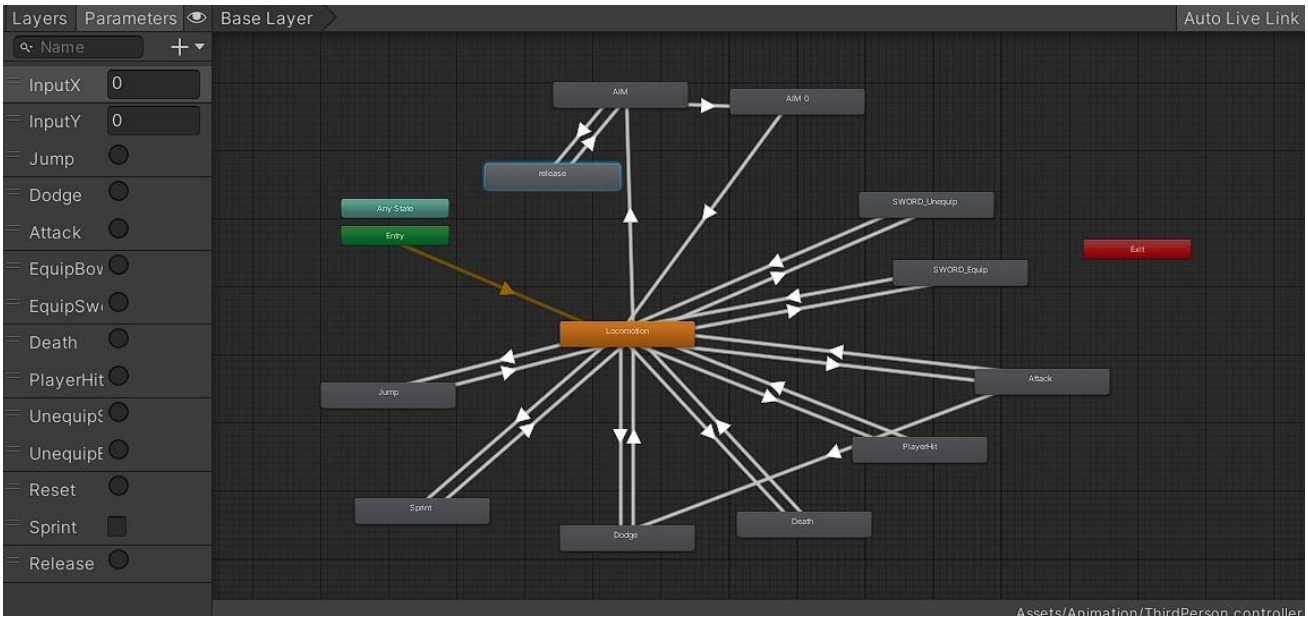

Figure 56: Player's Animator

The image above is the animator used for the main character with all its parameters.

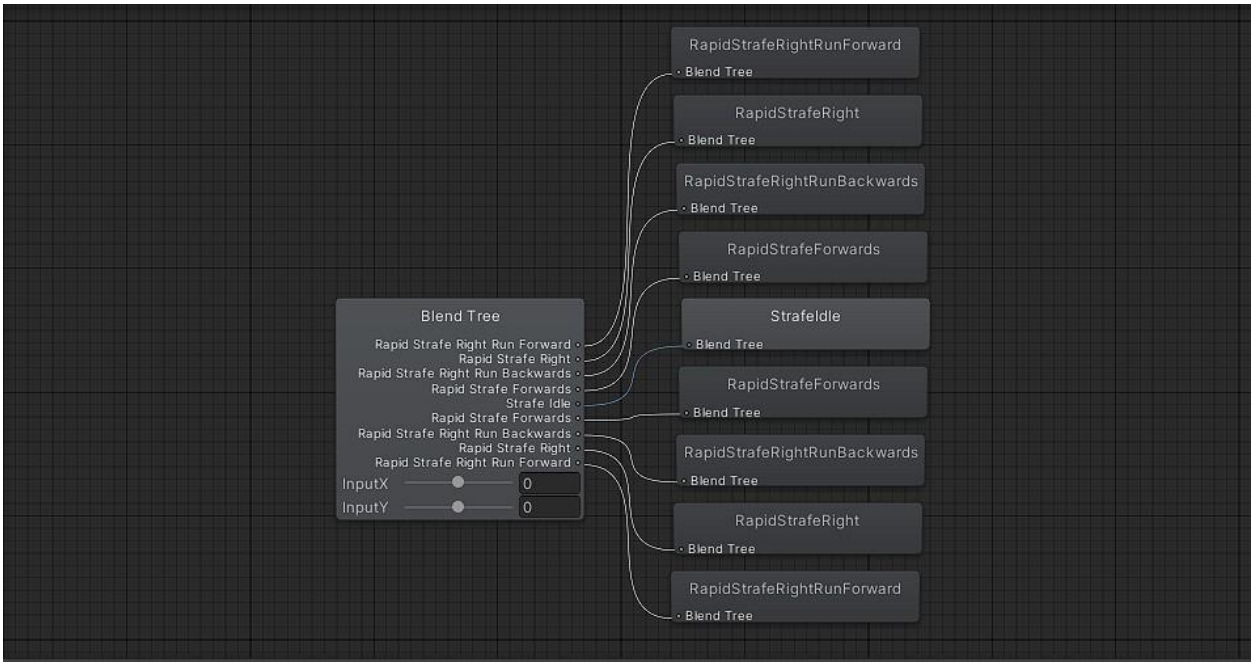

Figure 57: Player's Animator Locomotion Blend Tree

The image above is a blend tree of the base layer of the player's animator. It uses 2D Freeform Directional with parameters InputX and InputY.

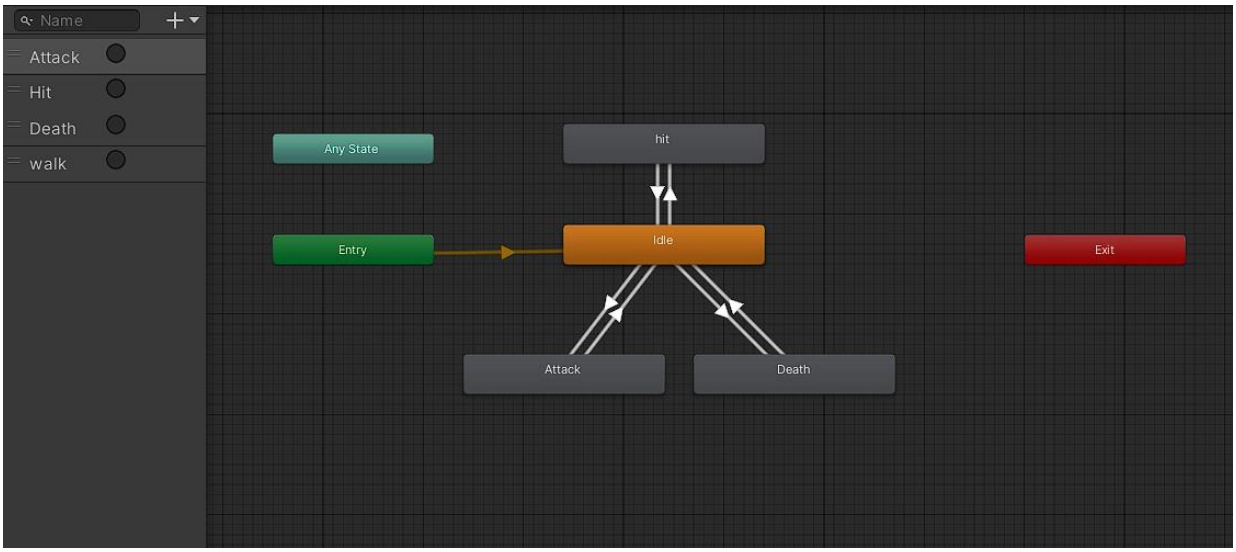

Figure 58: Default Mob Animator Locomotion Blend Tree

The Animator above is used for all the units in the game, excluding the player. It serves as the template for how the animations are played for the unit's behavior.

Post-Production

After the game has been tested, feedback has been provided and fixed during post-production. Completely the game has been uploaded to cloud storage, and links are sent out for the game to be downloaded for beta testing.

Usability Testing

The study is managed within Google Forms with specific respondents within the age range of 22 - 24 years old with an average age of twenty-three (23). The questions are used to comply with the concepts and themes of the game. When the respondents finish the game, they are requested to answer a survey by choosing one of the options on the six-point Likert scale. The developers surveyed ten selected respondents to complete their participation and enjoyment when playing *Unmask*. Due to time constraints, there is a limited number of respondents.

Gender	Age			Total
	22	23	24	
Male	1	4	1	6

Female	2	2	0	4
Total	3	6	1	10

Table 1: Unmask Respondents

With the data gathered by the developers, ten respondents participated in the survey due to time constraints. Of the respondents, with those aged 22-24, the game is more appealing to the male audience.

The developers use Game User Experience Satisfaction Scale (GUESS) for the survey. It is a scale verified by psychometricians that calculates user satisfaction by applying implemented sub-scales.

There are nine (9) sub-scales used in GUESS; under these sub-scales are fifty-five (55) statements. These nine sub-scales include Usability/Playability, Narratives, Play Engrossment, Enjoyment, Creative Freedom, Audio Aesthetics, Personal Gratification, Social Connectivity, and Visual Aesthetics. Using a 6-point Likert scale where one (1) is determined as Strongly Disagree while six (6) is determined as Strongly Agree.

Out of 55 statements within the sub-scales, the developers used 21 statements under the sub-scales that were necessary to apply to the game. Meanwhile, 1 statement falls outside the sub-scales of GUESS. The table below shows how many statements are present within each sub-scale in the survey. For more accurate reference, refer to Chapter 5: Appendices D. Survey Details.

Sub-scale	Number of Statements

Usability/Playability	5
Narratives	1
Play Engrossment	3
Enjoyment	4
Creative Freedom	1
Audio Aesthetics	3
Personal Gratification	1
Social Connectivity	0
Visual Aesthetics	3
Total	21

Table 2: Number of Statements for Each Sub-scale

According to the survey's responses, most respondents' gaming experience had fair results. The enjoyment accompanied by the audio aesthetics displayed excellent outcomes.

For the results' interpretation, the developers use the Top 2 Box as a proper tool from the Likert Scale questionnaires to get the satisfaction level per subscale.

The tool summarizes the percentage of respondents who agreed with the statement. To calculate the score for each dimension or subscale, the average of the scores for all statements below each subscale and calculate the GUESS score for the game as the average score for all subscales or dimensions.

Through the tool, the percentage of respondents who agreed with the statements is summarized. To determine the score per subscale, take the average scores for all statements below per subscale and calculate the GUESS score for the game as the average score for the entire subscale. The other 1 statement that is outside of the subscales is included after the nine subscales.

Subscale/Others	Top 2 Box	Percent Agree
Usability/Playability	94%	100%
Narratives	80%	100%
Play Engrossment	93%	97%
Enjoyment	95%	100%
Creative Freedom	80%	100%

Audio Aesthetics	100%	100%
Personal Gratification	100%	100%
Social Connectivity	0%	0%
Visual Aesthetics	93%	100%
Outside the Subscale	70%	80%
Overall Rating	81%	88%

Table 3: Summary of the Ratings of All Subscales/Others

Generally, this table presents that the respondents agree that Unmask's gameplay, visuals, and fresh storyline were admirable.

The table above displays the overall rating of the respondents about the game that belongs to the shown criteria.

Unmask has the highest ratings in Audio Aesthetics and Personal Gratification. The game features different game modes per chapter or level that let the respondents expect something new in progressing through the game.

For enjoyment, it has a high rating due to the previous statement of the game having a different game mode for each level.

For Usability/Playability, it also has a high rating because Unmask has a play style that is straightforward to understand prompts with textual guides as the players play along.

Both Play Engrossment and Visual Aesthetics have high ratings, as well as the correlation of the visuals in enhancing the gaming experience.

Narratives and Creative Freedom have a pretty good rating. With its uniqueness in terms of a unique storyline, Unmask features animated infographic cutscenes to define the game's plot further.

Outside the Subscale, it has a low yet decent rating. Understandably, the study's primary purpose is to raise awareness of a mostly unheard-of mental condition.

Lastly, Social Connectivity has a zero rating since only a single player plays the game.

CHAPTER IV

SUMMARY OF FINDINGS, CONCLUSIONS, RECOMMENDATIONS

Summary of Findings

The game was conceptualized during the pandemic. With everything unfamiliar to all, it was very overwhelming. Many changes were happening around the world, including the learning environment of educational institutions. Implementing an online setup lets others feel connected and disconnected simultaneously.

The developers then gathered ideas for a game that would be impactful and relevant to the present situation. It narrowed down to emotions, what most people feel, and a way of expression. Then they arrived at a topic that most had yet to hear. *Alexithymia* is a mental condition that has difficulty understanding or identifying emotions. Other studies suggest that people who are diagnosed with this condition, in other ways, display impulsive and uncontrollable emotions.

As to the development of the game, it is on a desktop platform due to the majority of the players use desktop computers or laptops. The key components that were emphasized were the masks of the characters. The masks are effective in portraying an expression or emotion. The developers went for a 3D approach to the game for it to be immersive and open many opportunities to explore. Then the developers laid out a linear storyline path that would pique the players' interest. Both correlate to the academic purpose of the study and the game's concept. And a 2D approach to the game's user interface, item icons, and cutscenes. These cutscenes include an intro, interaction, and

flashback cutscenes in an animated infographic video clip to explain the game's story further. The developers also made different game modes with variations of game mechanics per level within the game's four chapters. These were made to avoid repetitions that would bore the players. The assets are majority low poly for optimal game performance and a simplistic visual approach. The scripts were then coded for the games. The assets took the longest time to create since they took up almost all of the game's composition for it to come to a completion.

Conclusions

The game's primary purpose is to let players be aware and learn of a condition they probably did not know existed. Despite the condition, an average person could still struggle with a diagnosed individual's struggles. Through older generations, expressing emotions or feelings were mostly frowned upon and discouraged. However, times are changing, and the world is changing. The game is a potential eye-opener for others who feel it is normal. The game embeds a unique storyline per level that is based on personal experiences so the players can have a relative feel to it.

The challenge of the development of the game was scripting the AIs. Moreover, the developers could only survey a few respondents due to time constraints and circumstances. Despite the limited number of respondents, the collected data shows a reasonably good result of the game's purpose: to give awareness of Alexithymia.

Recommendations and Future Work

The developers recommend a more additional storyline or a non-linear storyline approach for more gaming exploration and possibilities of how the game progresses.

The developers also recommend changing the state of the interactable to help players recognize its change. Enemy AIs' behavior needs to be hostile for it to be reasonable. The death system should be more punishable by resetting the player's progress to the first chapter or level of the game.

The developers also suggest changing the game to online or cooperative to affect the Social Connectivity subscale of the Game User Experience Satisfaction Scale (GUESS).

Lastly, it would be nice to have more emotions to introduce into the game. This scale would cover a broader range of the study.

CHAPTER V

BIBLIOGRAPHY

[1] Granic, I., Lobel, A., & Engels, R. C. M. E. (2014). *The benefits of playing video games. American Psychologist*, 69(1), 66–78. https://doi.org/10.1037/a0034857

[2] Sophie Gaetan, Vincent Bréjard, Agnès Bonnet. *Video games in adolescence and emotional functioning: Emotion regulation, emotion intensity, emotion expression, and alexithymia.*

Computers in Human Behavior, 2016, 61, pp.344-349. ff10.1016/j.chb.2016.03.027ff. Ffhal-01314720f

[3] Simon Greipl, Korbinian Moeller, Manuel Ninaus.(2020). *Potential and limits of game-based learning* https://doi.org/10.1504/IJTEL.2020.110047

[4] Jens Kjeldgaard-Christiansen (2020). *Narrative video game aesthetics and egocentric ethics.* https://doi.org/10.7146/mediekultur.v36i68.118777

[5] Weiss, T. C. (2022, August 23). *Alexithymia: Information, Symptoms and Treatment Options*. Disabled World. https://www.disabled-world.com/health/neurology/alexithymia.php

[6] *Florence on Steam*. (n.d.). (2020). Retrieved January 18, 2023, from Steam.: https://store.steampowered.com/app/1102130/Florence/

[7] *Actual Sunlight on Steam*. (n.d.). (2014). Retrieved January 18, 2023, from Steam.: https://store.steampowered.com/app/288040/Actual_Sunlight/

[8] *Hidden Folks on Steam*. (n.d.) (2017). Retrieved January 18, 2023, from Steam: https://store.steampowered.com/app/435400/Hidden_Folks/

[9] Wikipedia contributors. (2023, January 8). *That Dragon, Cancer*. Retrieved January 18, 2023, from Wikipedia: https://en.wikipedia.org/wiki/That_Dragon,_Cancer

[10] *Night in the Woods on Steam*. (n.d.). (2017). Retrieved January 18, 2023, from Steam: https://store.steampowered.com/app/481510/Night_in_the_Woods/

[11] Blender Foundation. (2022). *About*. blender.org. Retrieved January 18, 2023, from Blender: https://www.blender.org/about/

[12] *Clip Studio Paint*. (2022). CLIP STUDIO.NET. Retrieved January 18, 2023, from Clip Studio Paint: https://www.clipstudio.net/en

[13] *Official Adobe Photoshop | Photo and design software*. (2022). Retrieved January 18, 2023, from Adobe: https://www.adobe.com/ph_en/products/photoshop.html

[14] *Professional video editing software | Adobe Premiere Pro*. (2022). Retrieved January 18, 2023, from Adobe: https://www.adobe.com/ph_en/products/premiere.html

[15]Technologies, U. (2022). *Real-time tools for 3D, AR, & VR development | Products*. Unity. Retrieved January 18, 2023, from Unity: https://unity.com/products

[16]Microsoft. (2022). *Visual Studio 2022 | Download for free*. Visual Studio. Retrieved January 18, 2023, from Visual Studio: https://visualstudio.microsoft.com/vs/

APPENDICES Development Tools

https://unity.com

https://visualstudio.microsoft.com

Clip Studio Paint

Multimedia

Blender

Tools

https://www.clipstudio.net/

https://www.blender.org/

Adobe Photoshop 2020

Adobe Premiere Pro

https://www.adobe.com/products/photoshop.html https://www.adobe.com/products/premiere.html

Sample Cutscenes

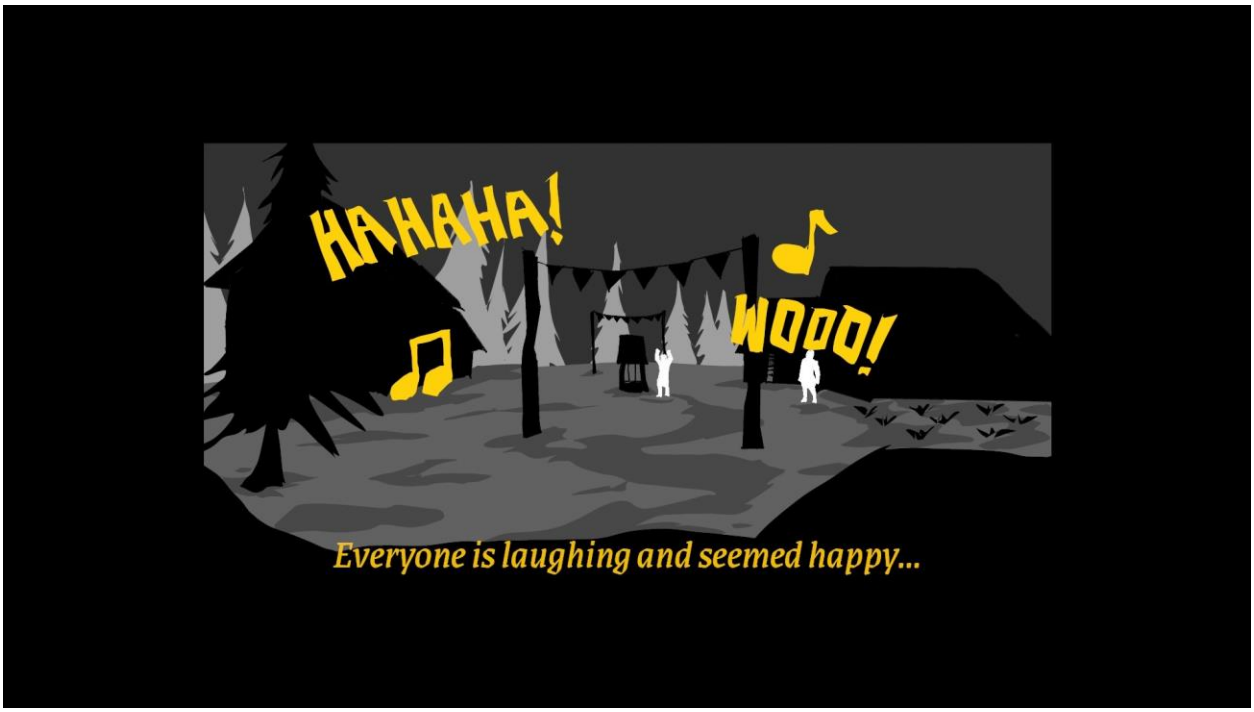

Figure 59: Sample Intro Cutscene

The image above is a sample intro cutscene of the fourth chapter. Intro cutscenes usually start the chapter or level of the game.

Figure 60: Sample Interaction Cutscene

The image presented above is the interaction cutscene between the player and the third-chapter boss. Interaction cutscenes usually appear after defeating each boss per level.

Figure 61: Sample Flashback Cutscene

The image above shows the flashback story of the boss in the first chapter.

Flashback cutscenes usually appear at the very end of a chapter. This is an indication for the player to move into the next chapter or level.

Survey Details

The tables below show the rating for each sub-scale:

Usability/Playability	Top 2 Box	Percent Agree
I think it is easy to learn how to play the game.	100%	100%
I find the controls of the game to be straightforward.	90%	100%
I do not need to go through a lengthy tutorial or read a manual to play the game.	90%	100%
I find the game's menus to be user friendly.	100%	100%
I feel the game provides me the necessary information to accomplish a goal within the game.	90%	100%
Average	94%	100%

Table 4: Usability/Playability

Narratives	Top 2 Box	Percent Agree
I think it is easy to learn how to play the game.	80%	100%
Average	80%	100%

Table 5: Narratives

Play Engrossment	Top 2 Box	Percent Agree
I feel the game constantly motivates me to proceed further to the next stage or level.	100%	100%
Whenever I stopped playing the game I cannot wait to start playing it again.	90%	90%
I can block out most other distractions when playing the game.	90%	100%
Average	93%	97%

Table 6: Play Engrossment

Enjoyment	Top 2 Box	Percent Agree
I enjoy the quests provided by the game.	80%	100%
I recommend this game to others.	100%	100%
If given the chance, I want to play this game again.	100%	100%
I enjoy playing the game.	100%	100%
Average	95%	100%

Table 7: Enjoyment

Creative Freedom	Top 2 Box	Percent Agree
I think the game is unique or original.	80%	100%
Average	80%	100%

Table 8: Creative Freedom

Audio Aesthetics	Top 2 Box	Percent Agree
I enjoy the sound effects in the game.	100%	100%
Whenever I stopped playing the game I cannot wait to start playing it again.	100%	100%
I can block out most other distractions when playing the game.	100%	100%
Average	100%	100%

Table 9: Audio Aesthetics

Personal Gratification	Top 2 Box	Percent Agree
I feel successful when I overcome the obstacles in the game.	100%	100%

Average	100%	100%

Table 10: Personal Gratification

Visual Aesthetics	**Top 2 Box**	**Percent Agree**
I enjoy the game's graphics.	100%	100%
I think the game is visually appealing.	100%	100%
I think the graphics of the game fit the mood or style of the game.	80%	100%
Average	93%	100%

Table 11: Visual Aesthetics

Outside the Sub-scales	**Top 2 Box**	**Percent Agree**
I'm now aware of the mental disorder Alexithymia.	70%	100%
Average	70%	80%

Table 12: Outside of the Sub-scales

Shown below is the sample survey form

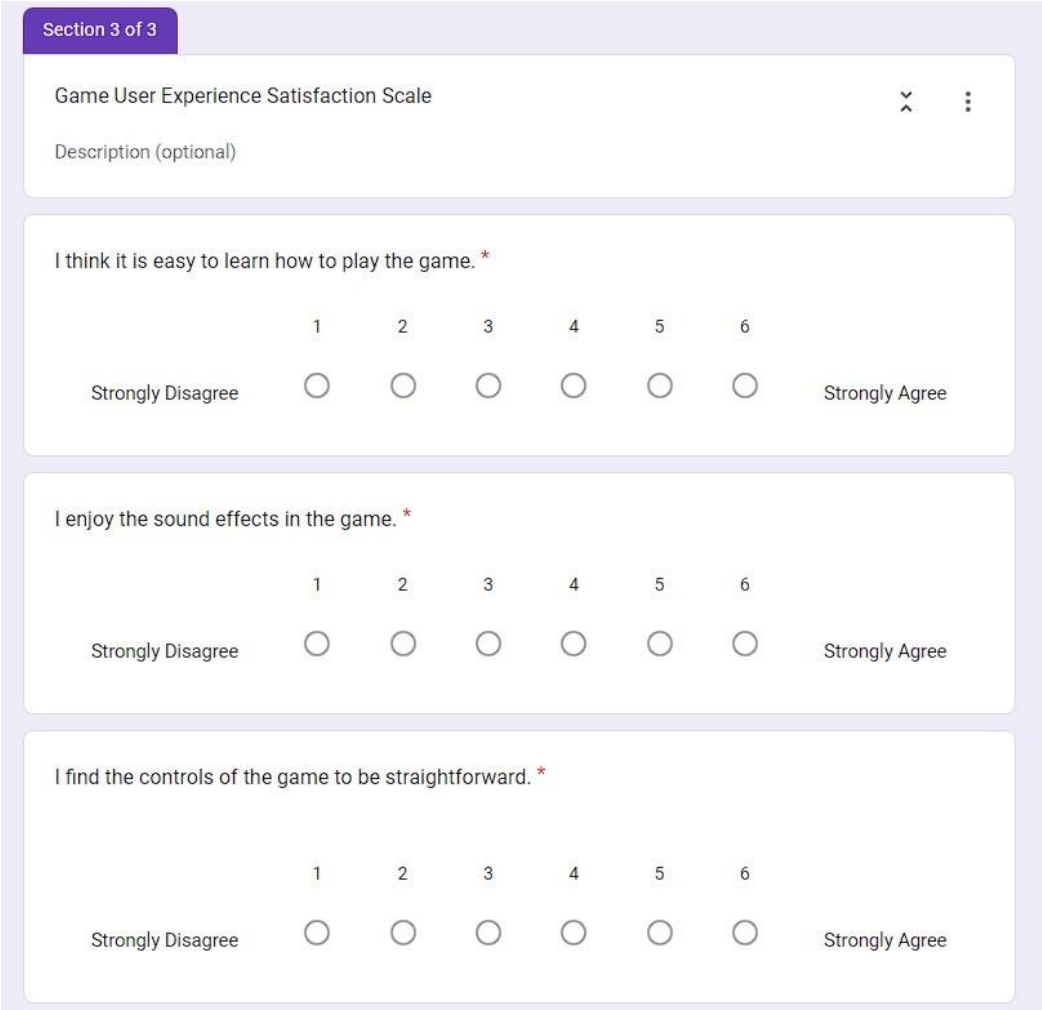

Figure 62: Sample Survey Form

Survey Respondents

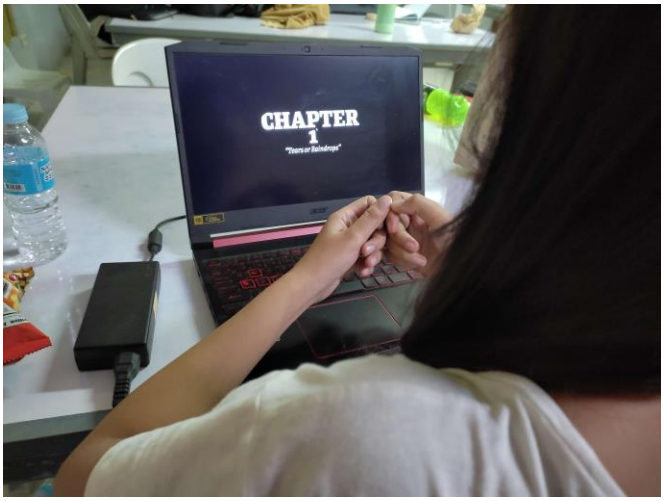

Figure 63: Respondent 1

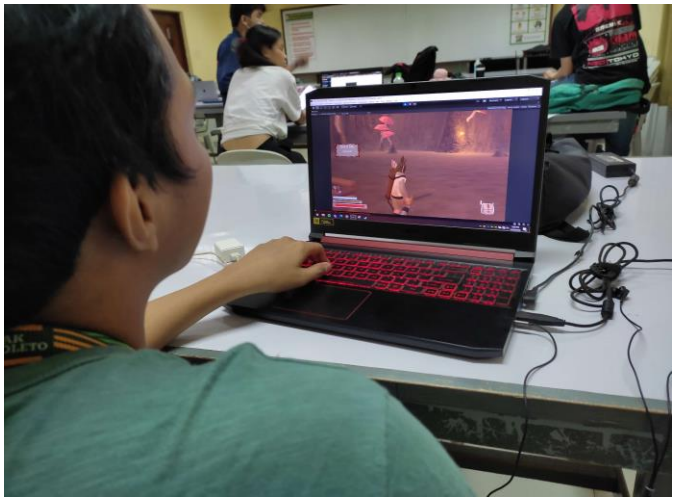

Figure 64: Respondent 2

Figure 65: Respondent 3

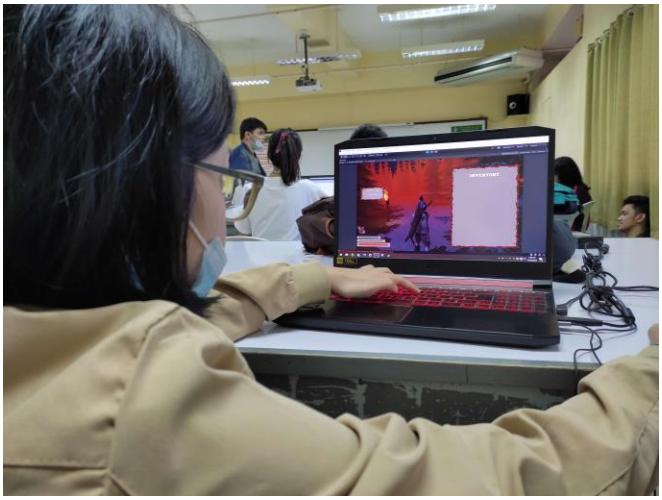

Figure 66: Respondent 4

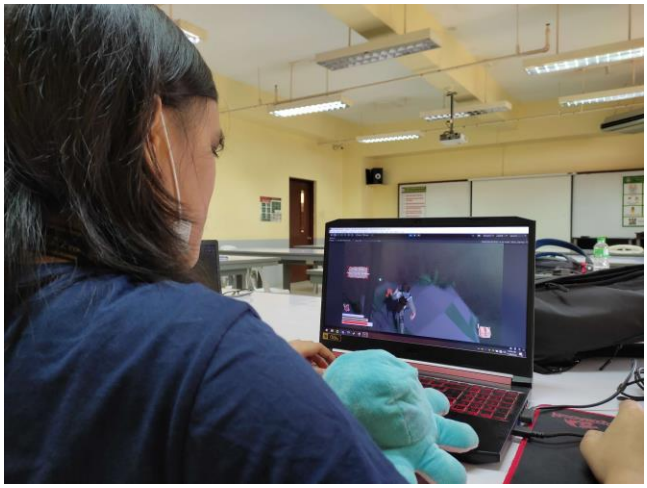

Figure 67: Respondent 5

Figure 68: Respondent 6

Figure 69: Respondent 7

Figure 70: Respondent 8

Figure 71: Respondent 9

Figure 72: Respondent 10

Asset Library

Building Assets

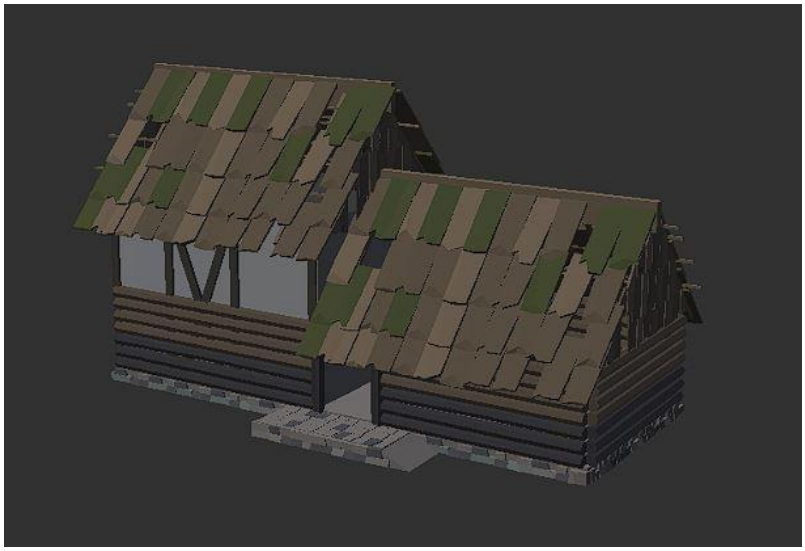

Figure 73: Villager House

Figure 74: Abandoned Tavern

Figure 75: Torn Down House

Terrain Assets

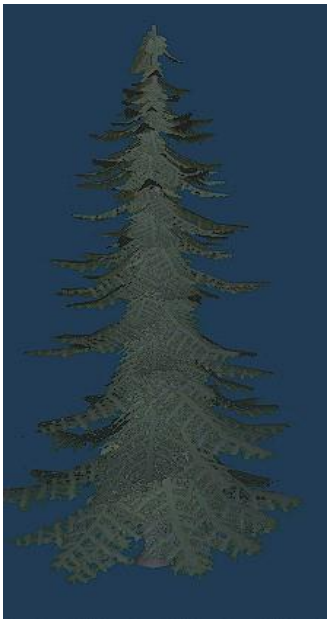

Figure 76: Pine Tree 1

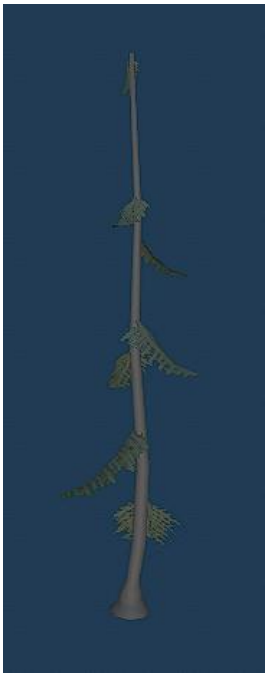

Figure 77: Pine Tree 2

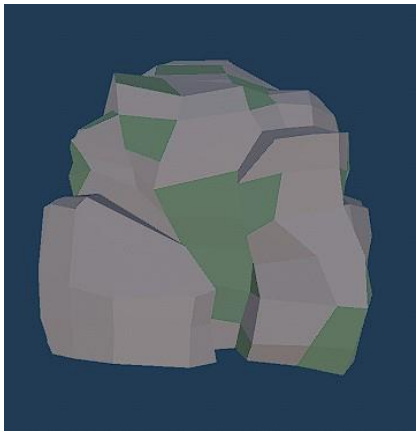

Figure 78: Mossy Rock 1

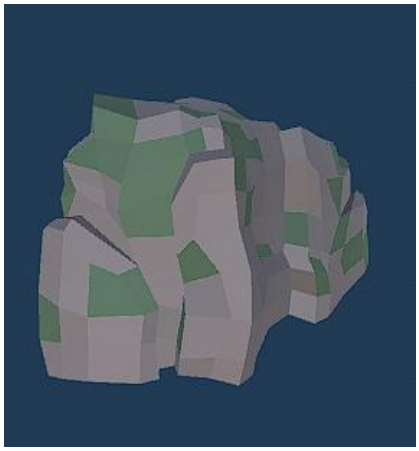

Figure 79: Mossy Rock 2

Figure 80: Mossy Rock 3

Figure 81: Dead Tree

Figure 82: Mushroom 1

Figure 83: Mushroom 2

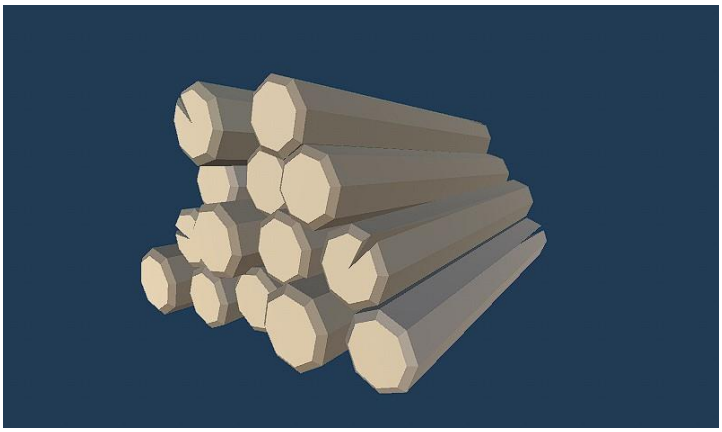

Figure 84: Pile of Firewood

Figure 85: Rack of Clothes

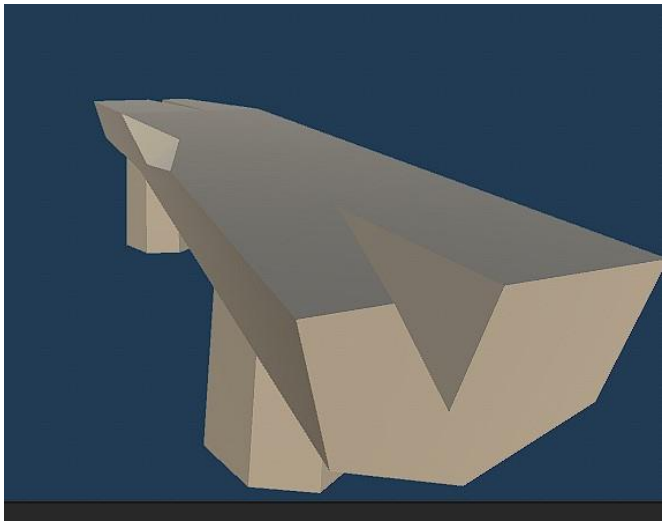

Figure 86: Wooden Bench

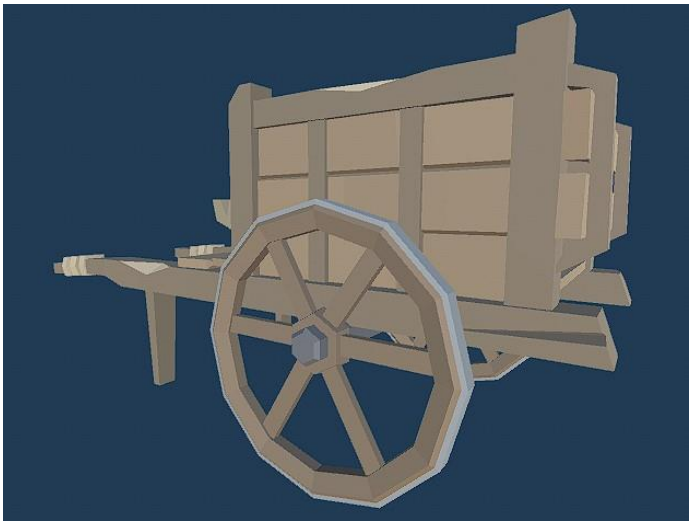

Figure 87: Wooden Cart

Figure 88: Village Well

Figure 89: Wooden Chair

Figure 90: Torch

Figure 91: Wooden Table

Interactables and Consumables

Figure 92: Wooden Sword

Figure 93: Iron Sword

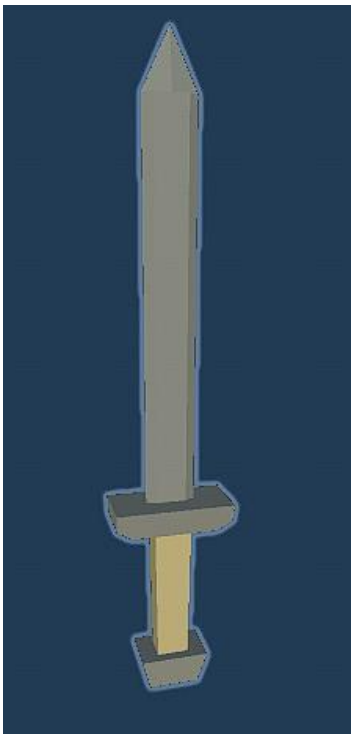

Figure 94: Steel Sword

Figure 95: Lever

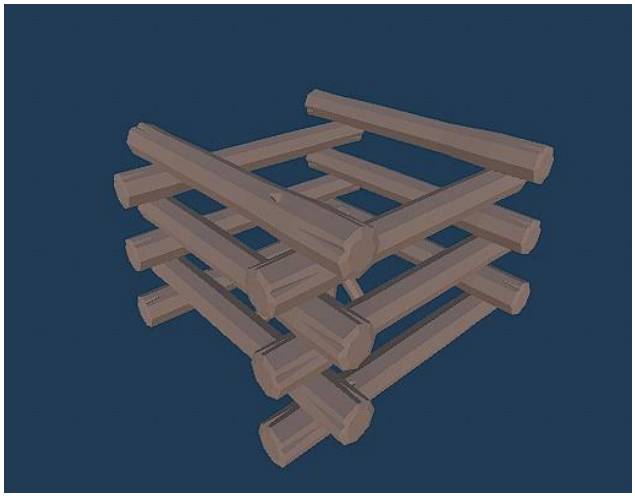

Figure 96: Bonfire Checkpoint

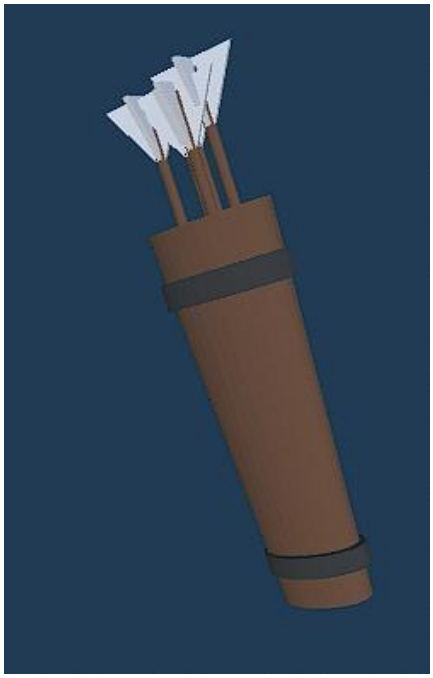

Figure 97: Arrows

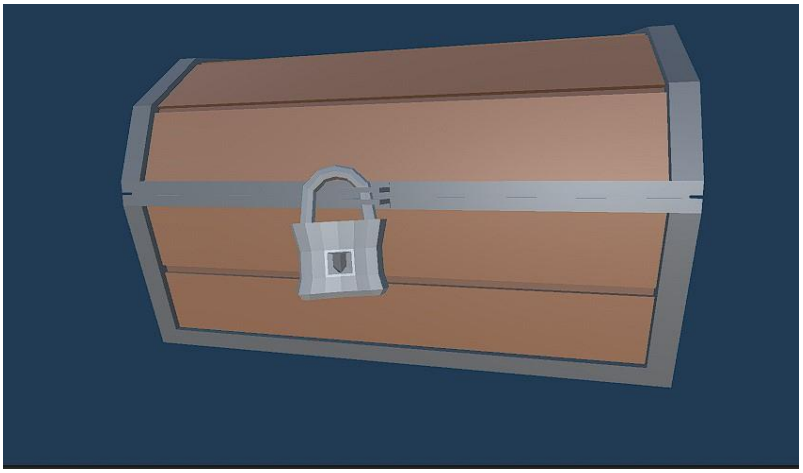

Figure 98: Chest

Figure 99: Leather Shield

Figure 100: Iron Shield

Figure 101: Steel Shield

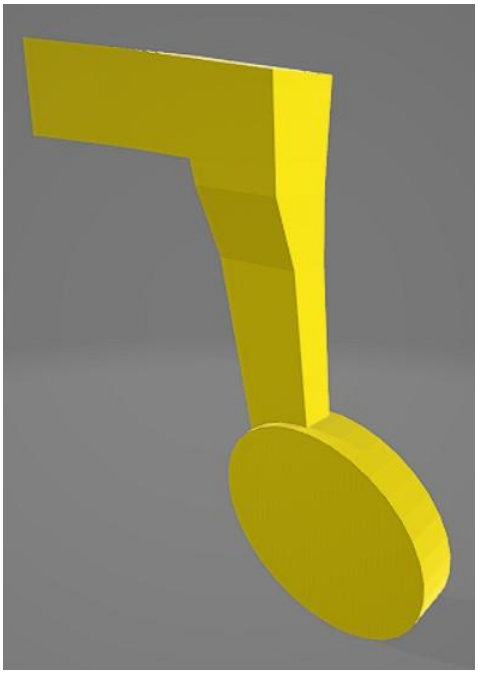

Figure 102: Music Note 1

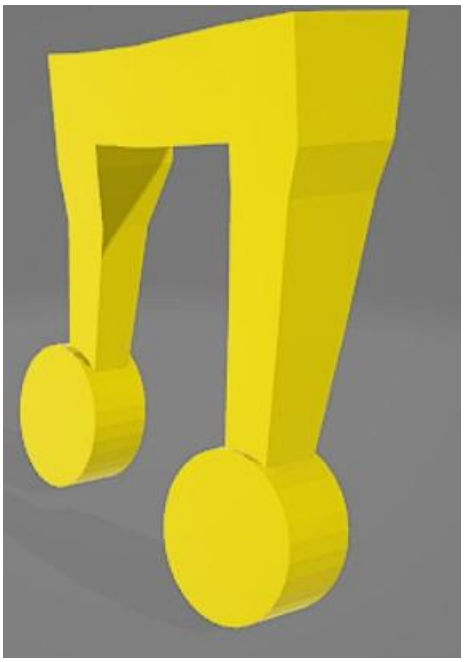

Figure 103: Music Note 2

Player, AIs, and Bosses

Figure 104: Player

Figure 105: Spider (Melee/Ranged)

Figure 106: Slime (Melee)

Figure 107: Slime (Ranged)

Figure 108: Bandit (Melee)

Figure 109: Bandit (Ranged)

Figure 110: Male Villager

Figure 111: Female Villager

Figure 112: Das / Sad

Figure 113: Reaf / Fear

Figure 114: Engar / Anger

Figure 115: Pyaph / Happy

SIMILARITY AND GRAMMARLY CERTIFICATE

University of San Jose - Recoletos
Corner P. Lopez & Magallanes Streets, 6000 Cebu City, Philippines
CENTER FOR POLICY, RESEARCH AND DEVELOPMENT STUDIES

CERTIFICATION

The manuscript entitled "**UNMASK: A THIRD-PERSON ROLE-PLAYING GAME AIMS TO RAISE SOCIAL AWARENESS FOR MENTAL HEALTH CONDITION ALEXITHYMIA**" has undergone Similarity and/or grammar tests using Turnitin and/or Grammarly software.

AUTHOR/s

Charles William D. Catingub; Joash Miguel G. Dandan; Niño Anthony L. Soldevilla

SIMILARITY INDEX: 8%

GRAMMARLY RESULT: 99/100

Given this 3/7/2023 15:15:34 at the Quality Assurance Unit of the Center for Policy, Research and Development Studies, University of San Jose-Recoletos, Cebu City. This certification is issued for **Research Publication** purposes.

MR. ENGLEVERT P. REYES, LPT, M.Ed, PhD(c)
CPRDS Director

OR #: 9007821357236

This certification is NOT VALID WITHOUT the Latest PDF RESULTS

CURRICULUM VITAE

Personal Information:

Name: Charles William D. Catingub

Address: Purok Mahogany 1 Tayud Liloan Cebu, 6002

Date of Birth: October 31, 1999

Contact Number: 09686829371

Nationality: Filipino

Email Address: cwcatingub@gmail.com

Academic Background:

Primary: Tayud Elementary School (2012)

Secondary: Tayud National High School (2012 - 2016)

University of Cebu Banilad Senior High School (2016 - 2018) **Tertiary:**

University of San Jose - Recoletos (2018 - 2022)

Professional Skills:

Strong: Unity 3D, C#, Blender, Adobe XD

Knowledgeable: C#, Adobe Photoshop, Adobe Illustrator, Blender.

Internship:

Pluck Games | Technical Game Designer

Personal Information:

Name: Joash Miguel G. Dandan

Address: Sitio Ylaya, Talamban, Cebu City

Date of Birth: October 24, 1999

Contact Number: +639272094125

Nationality: Filipino

Email Address: joashmigueldandan@gmail.com

Academic Background:

Primary: Maria Montessori International School - Talamban Campus (2003 - 2012)

Secondary: Academia de San Isidro Labrador (2012 - 2016)

University of Cebu Banilad Senior High School (2016 - 2018)

Tertiary: University of San Jose - Recoletos (2018 - 2022) **Professional**

Skills:

Strong: Adobe Photoshop, Adobe Premier, Adobe XD

Knowledgeable: Adobe Illustrator, Blender, Unity 3D, C#

Internship:

Pluck Games | Technical Game Designer

Personal Information:

Name: Niño Anthony L. Soldevilla

Address: Puntod Alaska Mambaling, Cebu City

Date of Birth: October 24, 1999

Contact Number: +9669209324

Nationality: Filipino

Email Address: soldevilla1231@gmail.com

Academic Background:

Primary: St. Mary's Academy Of San Nicolas (2003 - 2012)

Secondary: St. Mary's Academy Of San Nicolas (2012 - 2016)

St. Mary's Academy Of San Nicolas (2016 - 2018) **Tertiary:**

University of San Jose - Recoletos (2018 - 2022)

Professional Skills:

Strong: Blender, Unity 3D, C#

Knowledgeable: Blender, Unity 3D, C#

Internship:

Pluck Games | Technical Game Designer