



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Wrong Turn: A 3D Horror Game**  
**A PROJECT REPORT**

**Submitted to:**

**Department of Computer Application**

***Kathmandu Bernhardt College***

***In partial fulfillment of the requirements for the Bachelors in Computer  
Application***

**Submitted by:**

**Aman Bhandari**

**Registration no:**

**6-2-461-53-2018**

**Under the supervision of**

**Aanand Sah**

## ABSTRACT

Entertainment has always been a big part of life for everyone. Young or old everyone needs entertainment to enjoy their life and for their mental wellbeing. Wrong Turn is an app which is meant to provide entertainment. Wrong Turn is a horror game with an element of a scary environment, it is supposed to provide a scary experience for anyone who plays the game. Horror games are a special kind of entertainment which is sought out by many people. Being scared is a part of thrill that people would like to experience in their life. Wrong Turn is supposed to provide that thrilling experience to the players.

In the game "Wrong Turn," the creature uses the A\* algorithm to find the quickest way to get to a special disk that's placed in different spots in the game. A\* helps the creature in the game figure out how to move around efficiently. This makes the game more challenging for the player. The algorithm helps the creature decide the best paths to follow when they're chasing the playable character.

**Keywords:** A\* algorithm

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Project Supervisor **Mr. Aanand Sah**, he continuously helped and guided us through different problems faced during this whole coding process. **Wrong Turn** based on Unity Framework.

I would like to thank BCA Coordinator **Mr. Rambabu Mahato** for assisting in the different scope of the database of the programs. I would like to thank Kathmandu Bernhardt College for providing all the resources and ideal environment for the project developing process and encourage us to develop such a project.

I would also like to share our gratitude to Tribhuvan University for providing the opportunity to evaluate our programming skills through this project.

Aman Bhandari

# Table of Contents

<b>ABSTRACT .....</b>	<b>i</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>ii</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>v</b>
<b>LIST OF TABLES .....</b>	<b>vi</b>
<b>LIST OF FIGURES .....</b>	<b>vii</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Problem Statement .....	1
1.3 Objectives .....	1
1.4 Scope and Limitation .....	2
1.4.1 Scope .....	2
1.4.2 Limitation.....	2
1.5 Development Methodology.....	2
1.6 Report Organization .....	3
<b>CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW .....</b>	<b>5</b>
2.1 BACKGROUND STUDY .....	5
2.2 Literature Review.....	5
<b>CHAPTER 3: SYSTEM ANALYSIS AND DESIGN .....</b>	<b>7</b>
3.1 System analysis .....	7
3.1.1 Requirement Analysis.....	7
3.1.2 Feasibility Analysis .....	9
3.1.3 Object Modelling: Class Diagram .....	10
3.1.4 Dynamic Modelling: State & Sequence diagram.....	11
3.1.5 Process Modelling : Activity Diagram .....	12

3.2 System Design .....	13
3.2.1 Component Diagram.....	13
3.2.2 Deployment Diagram.....	14
3.3 Algorithm details.....	15
<b>CHAPTER 4: IMPLEMENTATION AND TESTING .....</b>	<b>17</b>
4.1 Implementation .....	17
4.1.1 Tools used.....	17
4.1.2 Implementation Details of Modules (Description of procedures) .....	19
4.2 Testing.....	20
4.2.1 Test Cases for Unit Testing .....	20
4.2.2 Test Cases For System Testing.....	21
<b>CHAPTER 5 : CONCLUSION AND FUTURE RECOMMENDATION.....</b>	<b>22</b>
5.1 Lesson learnt / Outcomes .....	22
5.2 Conclusion .....	22
5.3 Future recommendations.....	23
<b>REFERENCES.....</b>	<b>23</b>
<b>APPENDICES</b>	

## **LIST OF ABBREVIATIONS**

CASE	Computer Aided Software Engineering
FPS	Frames Per Second
GPU	Graphical Processing Unit
OOP	Object Oriented Programming
SQL	Structured Query Language
SDLC	Software Development Lifecycle
UML	Unified Modeling Language
UI	User Interface

## **LIST OF TABLES**

Table 1 : Unit Testing of Wrong Turn(A 3d Game) .....	20
Table 2 : System Testing of Wrong Turn (A 3d Game) .....	21

## LIST OF FIGURES

Figure 1 : Iterative Model of Wrong Turn (A 3D Horror Game) .....	3
Figure 2 : Use Case Diagram of Wrong Turn (A 3d Horror Game) .....	8
Figure 3 : Gantt Chart of Wrong Turn (A 3d Game) .....	9
Figure 4 : Class Diagram of Wrong Turn (A 3d Game) .....	10
Figure 5 : Sequence Diagram of Wrong Turn.....	11
Figure 6 : State Diagram of Wrong Turn (A 3d Game).....	12
Figure 7 : Activity diagram for Player Moment of Wrong Turn (A 3d Game) .....	13
Figure 8 : Component Diagram of Wrong Turn (A 3d Game) .....	14
Figure 9 : Deployment Diagram of Wrong Turn (A 3d Game) .....	14



# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

In the world of video games, there's a type of game that's really good at one feel jump scare and totally involved in the story. "Wrong Turn" is one of those games. It's a game where The character faces a world filled with terrifying creatures, unable to engage in combat, and must rely solely on their survival instincts to stay alive.

In the game "Wrong Turn," players assume the role of Luna, an intrepid journalist from Italy, who embarks on an investigative journey to a remote and eerie colony in Italy. Luna's insatiable curiosity drives her to uncover the unsettling mysteries of this isolated place. However, upon her arrival, she stumbles upon an unimaginable horror—a ravenous and relentless creature perpetually on the hunt for its next meal.

In the game "Wrong Turn," players are unable to confront the menacing adversaries, as attempting to do so results in failure. Victory in the game hinges on one's adeptness at evading the relentless, hunger-driven monsters. Players must either flee or seek refuge from these terrifying creatures. As they navigate this treacherous environment, they can also locate enigmatic disks concealed throughout the colony, which aid in unraveling the unsettling secrets concealed within this eerie locale.

Everyone enjoys having fun, whether they're young or old. Entertainment is a big part of life and helps people feel good mentally. "Wrong Turn" is a computer game made for entertainment. It's a scary game that creates a spooky atmosphere to give players an exciting experience. Some people really like scary games because feeling scared can be thrilling. "Wrong Turn" aims to give players that thrilling experience.

## 1.2 Problem Statement

Horror games have always been a big part of video gaming industry. But recent trends focus more on “action-horror” as opposed to survival horror, despite the overwhelming popularity. Developers are adding “jump scares” which are loud and sudden scares which are a cheap way to imitate horror in games. However these jump-scares are not well received by all audiences. Good horror is all about subtlety, about buildup, about providing a sense of discomfort. Jump scares, for the most part, do absolutely none of those; all they do is providing the player with an annoying shock, something that, if overused, becomes a

nuisance, not a scary thrill.

My hypothesis here is that horror games have become increasingly about the reaction of the person playing rather than the experience of the play itself. This has led to an increased focus on aesthetic design while gameplay itself has been neglected. Horror experiences are being built entirely around major set-pieces where the focus is only on the big-moments, with all the pieces in between acting in their service. This can lead to a pacing feeling fragmented.

### **1.3 Objectives**

The major objectives of wrong turn project are given below:

- To develop a third-person survival horror video game featuring journalist as the protagonist
- To implement gameplay mechanics that emphasize stealth and survival rather than combat.
- To implement the A\* algorithm to create intelligent enemy AI behavior.

### **1.4 Scope and Limitation**

#### **1.4.1 Scope**

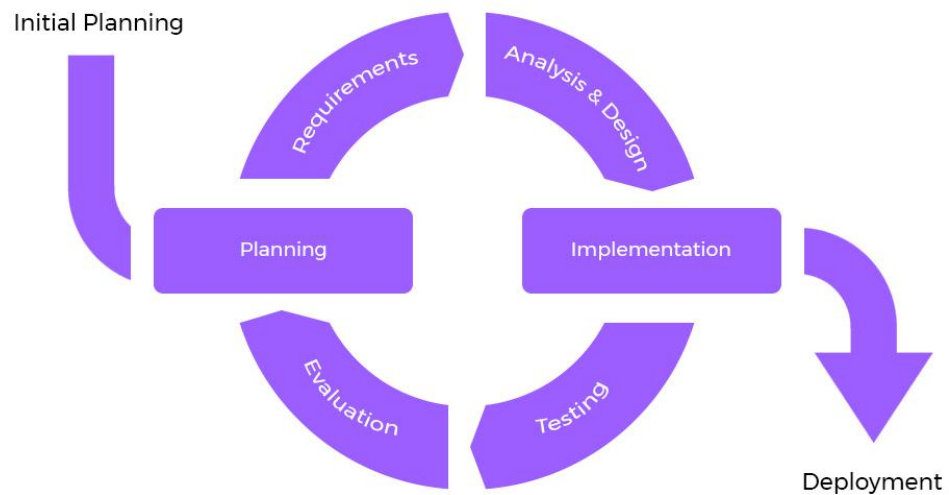
- It's designed to create a spooky atmosphere and make players feel scared and thrilled.
- The game is made to be fun, and it's for people who like scary and survival games.

#### **1.4.2 Limitation**

- The game supports linear approach where player follow a predetermined path.
- The absence of the ability to fight zombie.

### **1.5 Development Methodology**

The iterative model means doing things over and over to make them better when making a game. It's good for game-making because games change a lot, and also needs to keep fixing them to make them really fun.



**Figure 1 : Iterative Model of Wrong Turn (A 3D Horror Game)**

The iterative model is a systematic approach characterized by breaking down a project or process into a series of manageable cycles or iterations, each of which involves incremental development, feedback gathering, and adjustment, thereby fostering adaptability and continuous refinement, and it finds widespread application in diverse domains, including software development, product design, project management, scientific research, and complex problem-solving, allowing teams and individuals to respond to changing requirements and improve outcomes based on experience and insights gained during each iteration.

## **1.6 Report Organization**

The report on ‘Wrong Turn’ consists of five chapters.

**Chapter 1** discusses the need and features of **Wrong Turn** along with the problem statement and objectives of the project.

**Chapter 2** analyses the existing game along with background study and literature review of other horror games.

**Chapter 3** summarizes the game design along with the requirement analysis and feasibility analysis. The system design section encompasses various diagrams, such as the class diagram, object diagram, sequence diagram, activity diagram, and component diagram. These diagrams serve different purposes in illustrating the system’s structure, behavior,

and interactions.

**Chapter 4** explains the tools that are used on our project's unity framework, visual studio code, blender, photoshop, C#. The modules and the development model used are also explained. The unit testing and system testing along with the test cases performed is also explained in this part.

**Chapter 5** discusses the conclusion of how the project is accomplished, its findings, and many more. Different recommendation for future enhancements of the project is discussed here. In conclusion, this chapter overview's the purpose of doing this project including its scopes and objectives.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1 Background Study

One of the best-defined and most common types of horror games are survival horror games. These games tend to focus on the survival of the player-character in a horror setting with limited resources, and thus tend to be more geared as an action game or action-adventure game.<sup>[1]</sup> A common theme of these games is escape or survival from the equivalent of a zombie apocalypse, with weapons, ammunition, and armor limited. The Resident Evil series coined the term and serves as the prime example of such games, though key conventions of the subgenre preceded the Resident Evil series with games like 1992's *Alone in the Dark*.<sup>[2]</sup> Other notable survival horror series include *Clock Tower*, *Fatal Frame*, and *Parasite Eve*.

## 2.2 Literature Review

**Layers of Fear** is a psychological horror video game developed by Bloober Team and published by Aspyr. It was released on Linux, Microsoft Windows, macOS, PlayStation 4, and Xbox One worldwide in February 2016.

In *Layers of Fear*, the player controls a psychologically disturbed painter who is trying to complete his magnum opus as he navigates a Victorian mansion revealing secrets about his past. The gameplay, presented in first-person perspective, is story-driven and revolves around puzzle-solving and exploration. *Layers of Fear: Inheritance* was released on 2 August 2016 as a direct follow up add-on to the first game. This time the player controls the painter's daughter with the downloadable content focusing on her apparent relapse into trauma after returning to her old house.

**Phasmophobia** is an investigative horror game developed and published by British indie game studio, Kinetic Games. The game received a large influx of popularity the following month due to many well-known Twitch streamers and YouTubers playing it, mainly for the Halloween season.<sup>[3]</sup> As of October 15, 2020, the game was the sixth-most popular game on Twitch.<sup>[4]</sup> It was the best-selling game on Steam globally for several weeks in October to November 2020. *Phasmophobia* is a horror investigation survival game played from a first-person perspective. The player works solo or in a group of up to four players to complete a contract in which they must identify the type of ghost haunting the specified

site. Players can communicate through voice chat, both locally within a short distance and globally via walkie-talkies. Phasmophobia features speechrecognition<sup>[4][5]</sup> allowing certain pieces of equipment and even the ghost to hear players speaking and understand key words. Completing objectives and gathering evidence provide a payout; complete as much as possible to level up, unlock maps and difficulties, and earn money.

Players create or join a lobby where preparation for the contracts is done. Phasmophobia gives players the option to customize their appearance by picking between eight different paranormal investigators to distinguish between group members. Located at a whiteboard in the lobby, a contract location and difficulty (Amateur, Intermediate, Professional or Nightmare) is selected. All equipment is purchased at this board and selected to be loaded into the Van from which the group will operate out of upon arriving at the site.

As of June 2021, there are nine sites of various sizes: small, medium, or large. The small maps include four suburban-style houses and two farmhouses, the medium maps are a high school and prison, and the large map is an asylum. Potential future maps located at an apartment building, mansion, sewer, and hospital have been hinted at. In August 2021 an update featured one small type map Willow Street house. An October 2021 update brought a medium sized outside type map named Maple Lodge Campsite, the first map in the game to be primarily set outdoors.

**Outlast** is a first-person survival horror video game developed and published by Red Barrels. The game revolves around a freelance investigative journalist, Miles Upshur, who decides to investigate a remote psychiatric hospital named Mount Massive Asylum, located deep in the mountains of Lake County, Colorado. The downloadable content Outlast was released for Microsoft Windows on September 4, 2013, PlayStation 4 on February 4, 2014 and for Xbox One on June 19, 2014. Linux and OS X versions were later released on March 31, 2015.<sup>[6]</sup> A Nintendo Switch version titled Outlast: Bundle of Terror was released in February 2018.

## CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

### 3.1 System analysis

When creating "Wrong Turn," system analysis is super important. It helps make the game better. How the game works now, like all its parts and how they fit together. A person called a system analyst looks at everything, talks to people, and asks questions. They want to find what's not working well and make it better. This way, it is understood how the game works, which is called the existing game system.

#### 3.1.1 Requirement Analysis

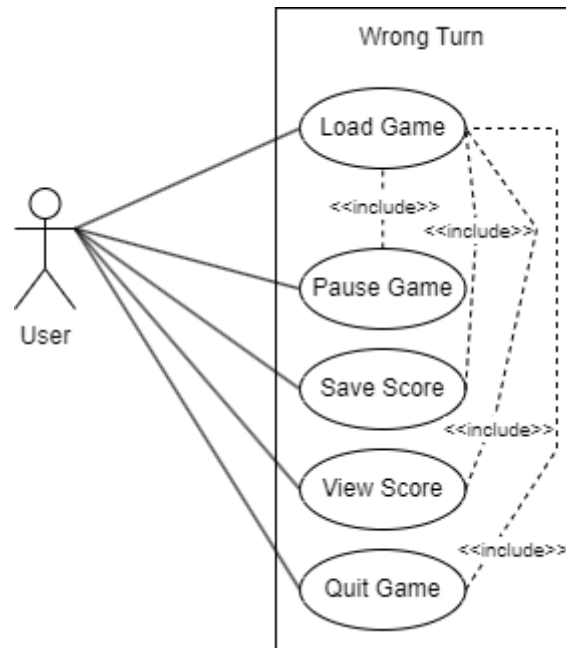
When making "Wrong Turn," requirement analysis is a very important step. It helps us figure out how the game should behave. How the game will connect with other parts and decide what the game must and must not do. This analysis has two main parts: functional, which is about what the game should do, and non-functional, which is about how well it should do those things.

##### i. Functional requirements

The functional requirements describe the system's behavior and the tasks it should perform. In the context of Wrong Turn, the functional requirements include:

- **Character control and movement:** In the game, Luna is a controllable and, the main character, with user-friendly and quick controls as she explores the mysterious world. Luna's movements will look and feel genuine, with animations for walking, running, and crouching that make the game more immersive.
- **Enemy Behavior:** In the game, the creature will be smart thanks to the A\* algorithm. They'll try to find Luna by looking and listening. These creatures won't always act the same - sometimes they'll be more aggressive, and other times they might back off, depending on what Luna does.
- **Collectible disk:** In the game, Luna can find special disks hidden in the colony, and collecting them helps her learn about the place's secrets. There's a counter that keeps track of how many disks she's collected, so players can see how well they're doing.
- **Game environment:** In the game, players will embark on an exploration of a haunting colony, immersing themselves in its eerie and foreboding atmosphere for a heightened

sense of realism. They will have the ability to interact with various elements within the environment, such as doors, windows, and hiding places. The game's interface will provide essential information, including Luna's energy levels, collected disks, and upcoming objectives, all seamlessly integrated to preserve the immersive and atmospheric experience without distracting from the game's ambiance.



**Figure 2 : Use Case Diagram of Wrong Turn (A 3D Horror Game)**

## ii. Non-functional requirements

The nonfunctional requirements of the wrong turn is given below:

- **Performance:** The game should run smoothly, maintaining a consistent frame rate of at least 60 frames per second on recommended computer systems. Additionally, seamless and rapid loading transitions between different sections of the game are crucial to ensure an enjoyable gaming experience.
- **Scalability:** The game is made in such a type where more stuff can be added to it later without making it slow or unstable. The environment and things in the game will be set up to work well on different types of computers.
- **Usability:** The game will feature user-friendly menus and provide helpful tips for newcomers. The control scheme will be designed to minimize frustration and enhance the immersion, allowing players to feel like they are truly inside the game.



- **Audio and Visual Quality:** A really good sounds and pictures to make the game feel super real and scary. The sounds, like things that go bump in the night and spooky music, will make the game even scarier.

### 3.1.2. Feasibility Analysis

#### a. Technical feasibility

The Project was technically feasible as it uses existing stable technology. The technology used includes Unity game engine, Adobe Photoshop, Blender which are within our technical capabilities.

#### b. Economic Feasibility

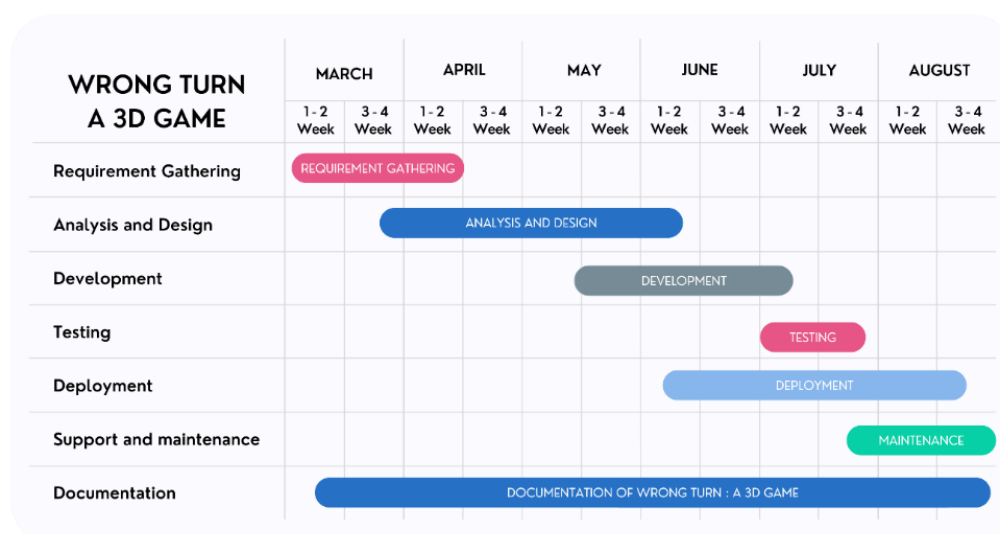
The project was economically feasible as all tools used to develop were completely free of cost for students.

#### c. Operational feasibility

The project is created using all the free tools and applications it will be available to all the users that are interested in games.

#### d. Schedule Feasibility

The project is developed within the time limit. Hence, it is feasible in the respective schedule. The below Gantt chart will show the mandatory deadlines that are expected to be achieved.



**Figure 3 : Gantt Chart of Wrong Turn (A 3D Horror Game)**

### 3.1.3 Object Modelling: Class Diagram

Class diagram represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. It shows a collection of classes, interfaces, associations, collaborations, and constraints.

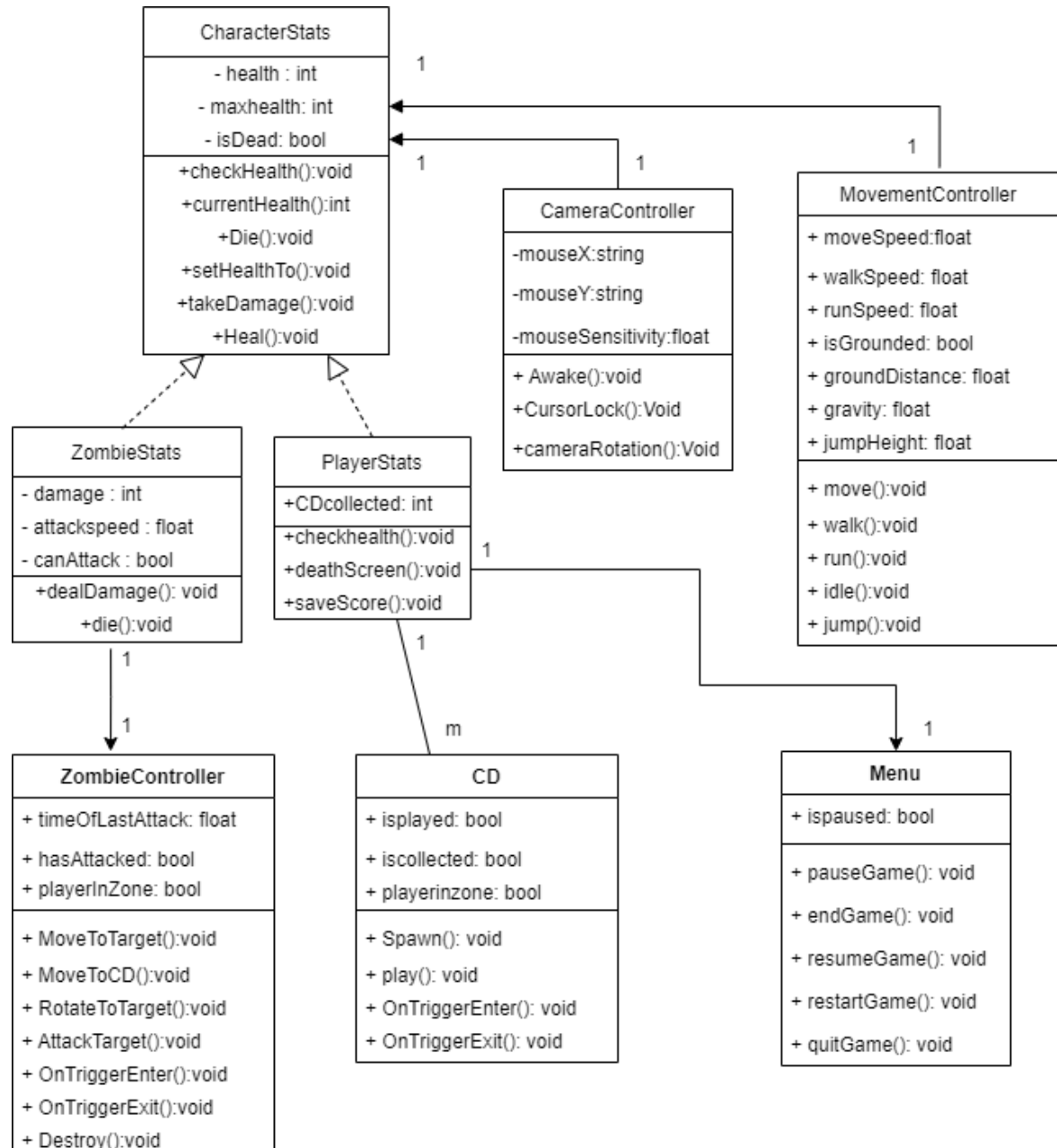


Figure 4 : Class Diagram of Wrong Turn (A 3D Horror Game)

### 3.14 Dynamic Modelling: State & Sequence diagram

A state diagram is a visual representation that illustrates the possible states, transition, and events of a system, aiding in modeling and understanding its behavior.

Sequence Diagrams are interaction diagrams that detail how operations are carried out. It describes interactions among classes in terms of an exchange of messages over time.

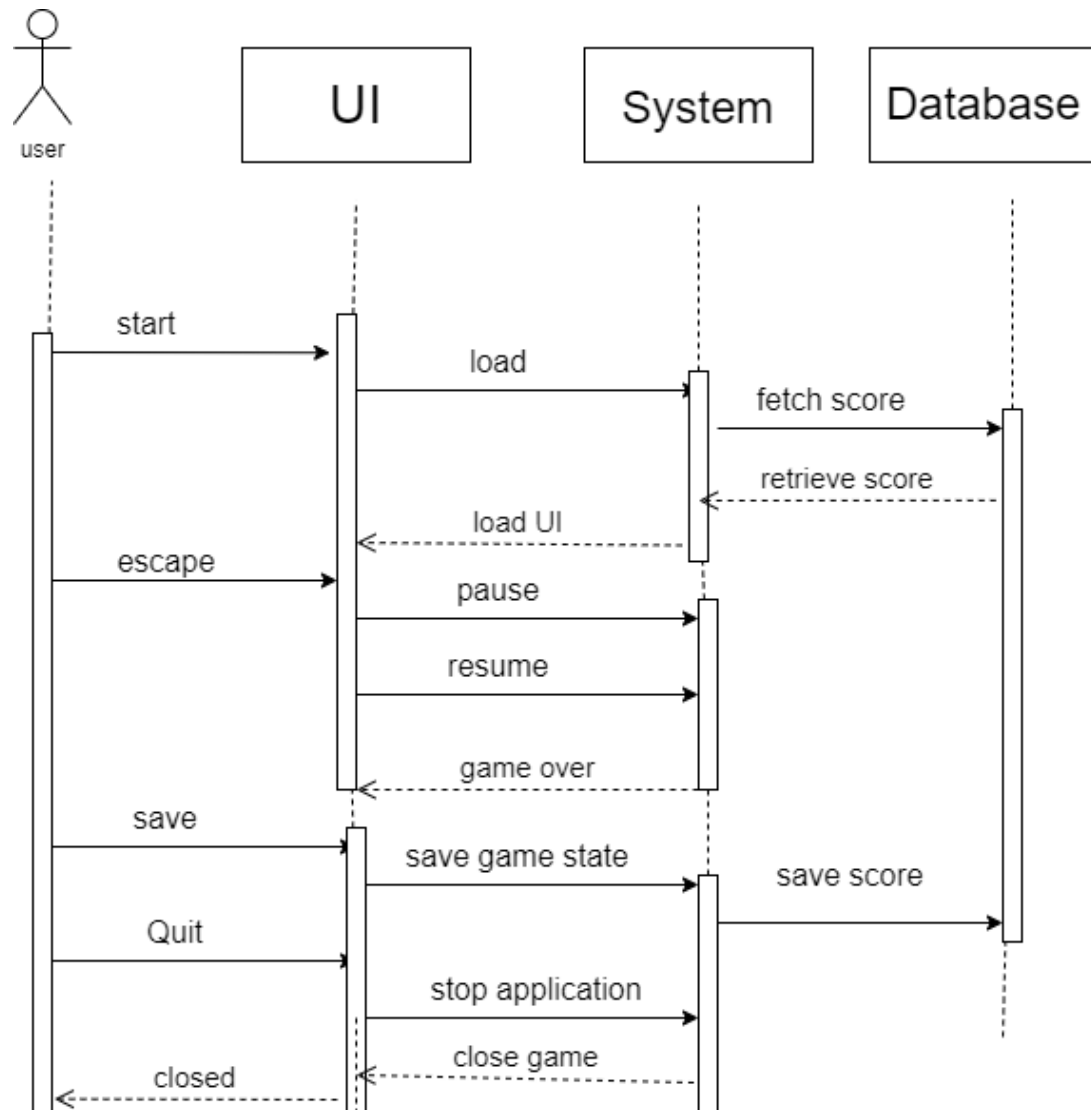
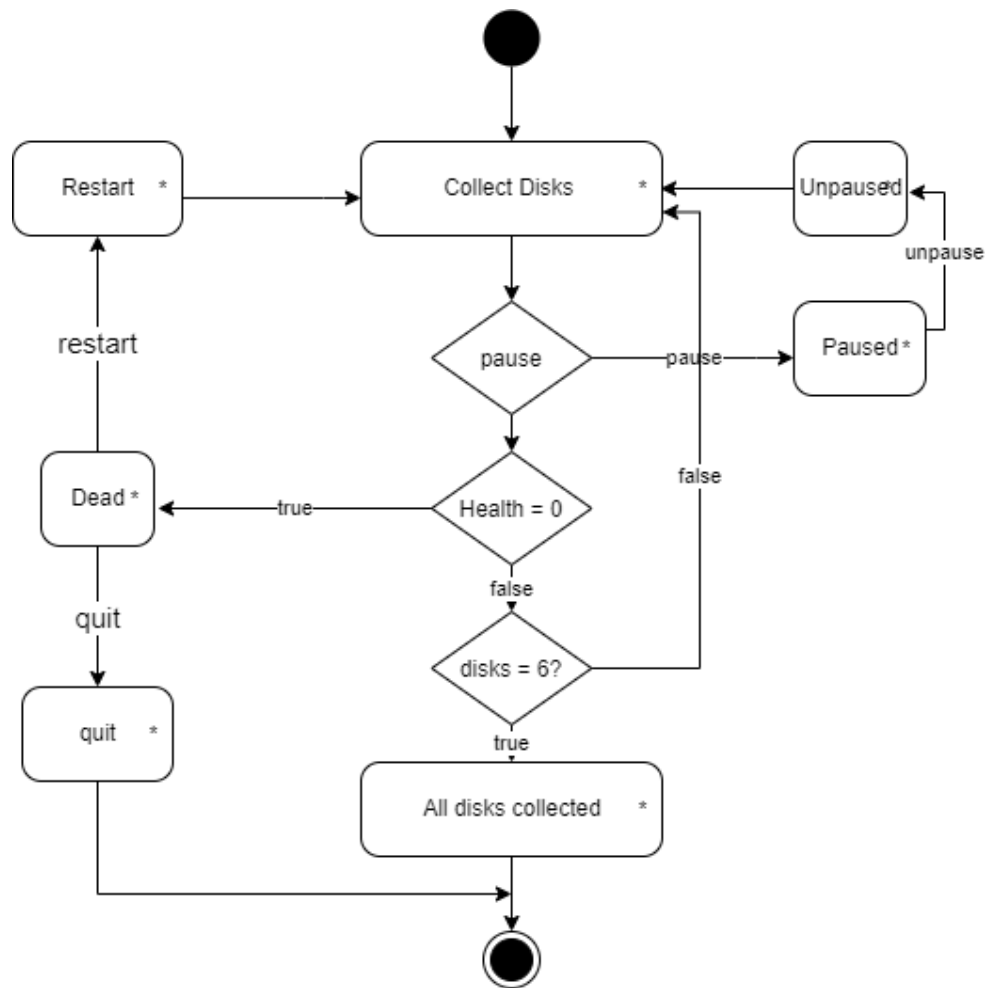


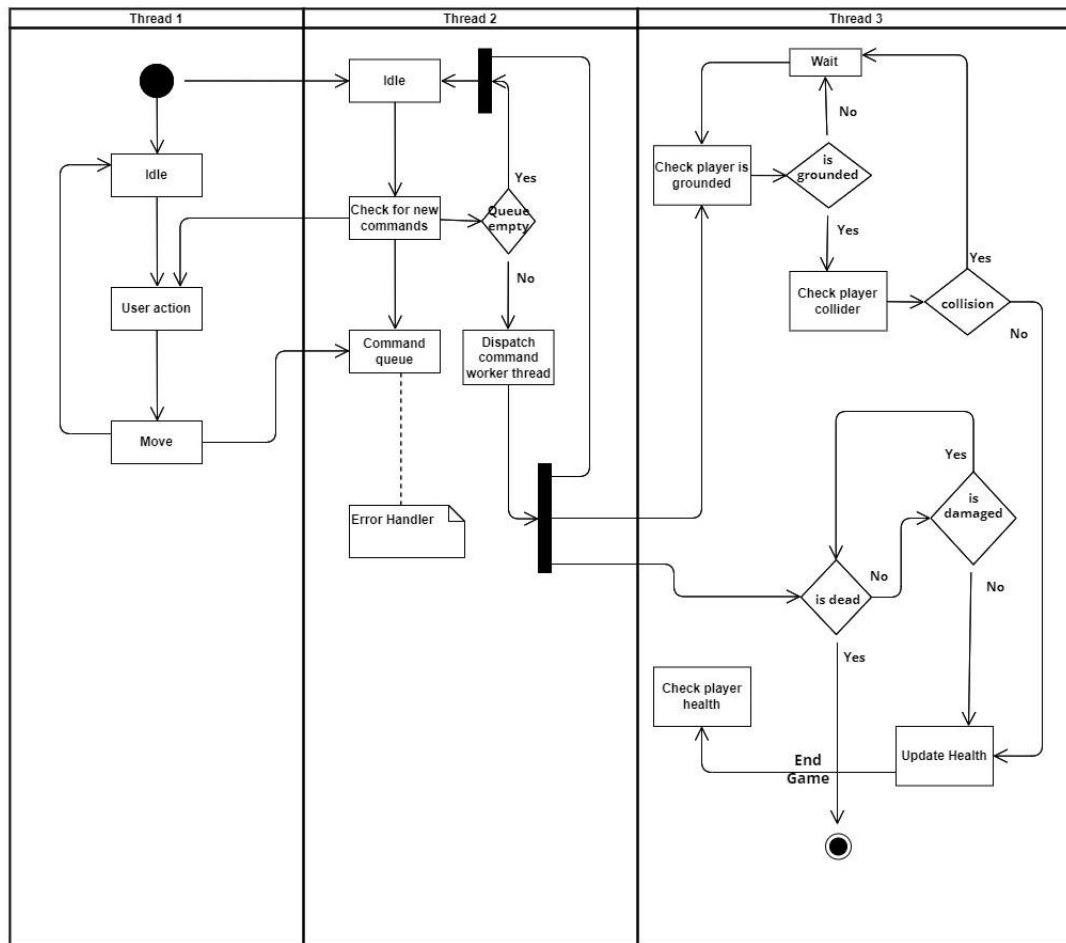
Figure 5 : Sequence Diagram of Wrong Turn (A 3D Horror Game)



**Figure 6 : State Diagram of Wrong Turn (A 3D Horror Game)**

### 3.1.5 PROCESS MODELLING: ACTIVITY DIAGRAM

The activity diagram is a flowchart to represent the flow of control among the activities in a system. It is another important diagram in UML to describe the dynamic aspects of the system. The activity can be described as an operation of the system. It consists of activities that are made up of smaller actions. It is an advancement of a flowchart that contains some unique capabilities.

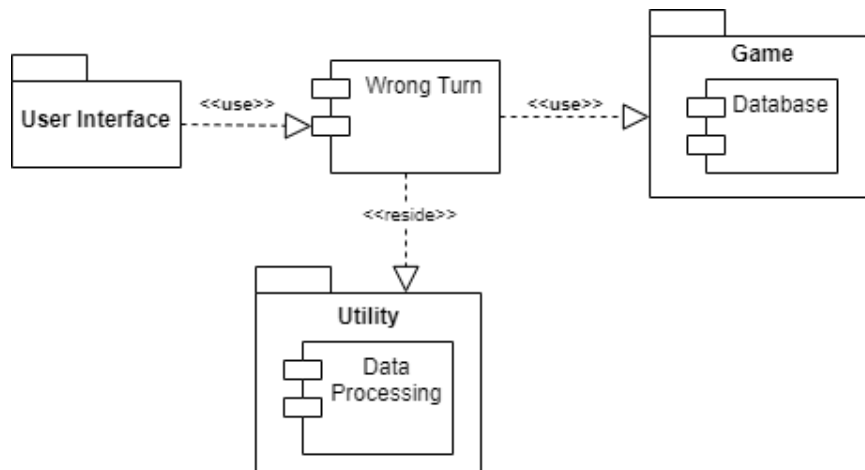


**Figure 7 : Activity diagram for Player Moment of Wrong Turn (A 3d Horror Game)**

## 3.2 System Design

### 3.2.1 Component Diagram

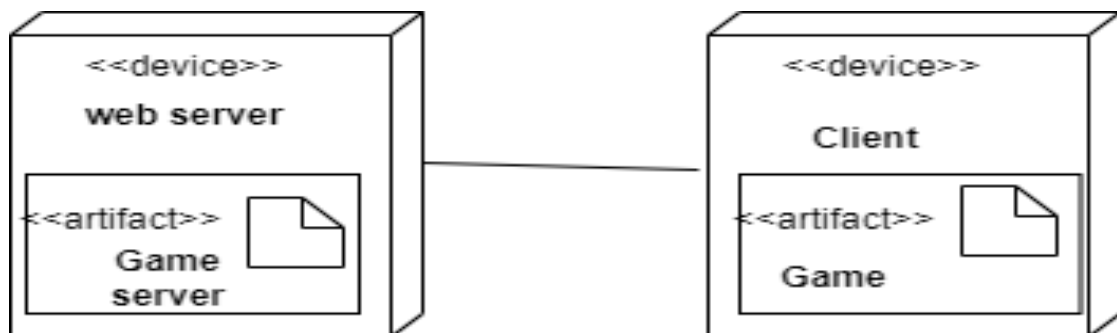
In the "Wrong Turn" game, a component diagram is like a picture that shows how all the parts of the game fit together. It helps us see how different things in the game connect to each other. This picture is useful when building the game and want to understand how it works. It helps us make sure that all the pieces of the game are put together correctly so that it can be played smoothly and be fun.



**Figure 8 : Component Diagram of Wrong Turn (A 3d Horror Game)**

### 3.2.2 Deployment Diagram

In the context of Unified Modeling Language, A deployment diagram is a diagram that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. It consists of nodes and their relationships.



**Figure 9 : Deployment Diagram of Wrong Turn (A 3d Horror Game)**

### 3.3 Algorithm details

A\* (pronounced "A-star") is a computer program trick that helps find the best way to get from one place to another on a map. It's used in computer science for lots of different things because it's really good at finding the best way to go and it does it pretty quickly. But it has a little problem - it uses a lot of computer memory to work. So, when using different routes on a computer, sometimes there are other ways that are faster and don't use as much memory. Still, A\* is really great for some problems, and it was made by some smart folks at Stanford Research Institute in 1968. It's like a smarter version of another program called Dijkstra's algorithm because it uses some smart shortcuts to find the best way to go.

#### **Working mechanism of A\* algorithm**

A\* is like a clever way to find the best route on a map. It looks at all the possible paths and picks the one that seems to be the shortest and most efficient. It does this by starting at one point, then adding more and more points to the route until it reaches the destination. At each step, A\* decides which path to take based on how far it's gone so far (called "g(n)") and a guess about how much farther it has to go (called "h(n)"). A\* keeps doing this until it either reaches the destination or can't find any more paths to try. The guess part ("h(n)") depends on the specific problem, but if it's a good guess and never overestimates, A\* will always find the best route from the start to the destination.

In the game "Wrong Turn," the creature use the A\* algorithm to find the quickest way to get to a special disk that's placed in different spots in the game. Usually, the disk is put near a creature that wants to grab it. When Luna, the main character, gets close to the creature, the creature tries to attack her. So, Luna has to run away from the creature and also try to grab the disk to finish the game.

A\* helps the creature in the game figure out how to move around efficiently. This makes the game more challenging for the player. The algorithm helps the creature decide the best paths to follow when they're chasing Luna and to travel near disk so he can grab the Luna.

```

private void MoveToTarget()
{
    agent.SetDestination(target.position);
    anim.SetFloat("Speed", 1f, 0.3f, Time.deltaTime);
    RotateToTarget();
    float distanceToTarget = Vector3.Distance(target.position, transform.position);
    if (distanceToTarget <= agent.stoppingDistance)
    {
        anim.SetFloat("Speed", 0f);
        if (!hasStopped)
        {
            hasStopped = true;
            timeOfLastAttack = Time.time;
            anim.SetTrigger("attack");
        }

        //Attack
        if (Time.time >= timeOfLastAttack + stats.attackSpeed)
        {
            timeOfLastAttack = Time.time;
            CharacterStats targetStats = target.GetComponent<CharacterStats>();
            AttackTarget(targetStats);
        }
    }
    else
    {
        if (hasStopped)
        {
            hasStopped = false;
        }
    }
}

```

```

private void MoveToCD()
{
    agent.SetDestination(CD.position);
    anim.SetFloat("Speed", 1f, 0.3f, Time.deltaTime);
    float distanceToTarget = Vector3.Distance(CD.position, transform.position);
    if (distanceToTarget <= agent.stoppingDistance)
    {
        anim.SetFloat("Speed", 0f);
        if (!hasStopped)
        {
            hasStopped = true;
        }
    }
    else
    {
        if (hasStopped)
        {
            hasStopped = false;
        }
    }
}

```



# **CHAPTER 4: IMPLEMENTATION AND TESTING**

## **4.1 Implementation**

### **4.1.1 Tools used**

#### **1. Unity framework**

Unity is an excellent choice for developing scary 3D games like "Wrong Turn" due to its user-friendly nature, making it accessible for creators without advanced skills. It offers a wide array of pre-made assets from its store, including eerie sounds and objects, streamlining the development process. Unity's tools enable realistic animations and movement for characters and objects, crucial for creating a chilling atmosphere, and provide control over character behaviors, such as fleeing from terrifying creatures. Additionally, Unity's lighting and special effects capabilities can contribute to the game's creepy ambiance. Furthermore, Unity's compatibility with various devices ensures a broader audience can enjoy the game. Nevertheless, proficient game design remains paramount in crafting an engaging and frightening gaming experience.

#### **2. C#**

C# is like a friendly and powerful tool. It's not too hard to learn, and it helps make the game work well. Unity and C# work together smoothly, so it's easy to tell the game what to do. C# in Unity is a good choice even to the new programmers. It can make the game have scary things happen, and it's good at making sure the game runs smoothly on different devices.

#### **3. Visual Studio Code**

Writing code in C# with VS Code offers valuable assistance in locating and rectifying errors, greatly aiding the coding process. Its flexibility enables developers to customize the code to suit their preferences and requirements. Additionally, its seamless integration with Unity makes it a powerful tool for game development, allowing creators to achieve the desired functionality and appearance in their games.

#### **4. Blender**

Blender is a valuable tool for creating 3D characters for games, offering a combination of being both free and powerful. It empowers creators to craft intricate characters with diverse appearances and enables realistic movement and behavior through its robust set of tools. With Blender, you can enhance characters by adding textures and colors, giving them a visually appealing and dynamic appearance. Its compatibility with popular game development platforms such as Unity and Unreal Engine further enhances its usefulness in the game development pipeline.

#### **5. Photoshop**

Adobe Photoshop is a raster graphics editor developed and published by Adobe Inc. for Windows and macOS. It was originally created in 1988 by Thomas and John Knoll. Since then, the software has become the industry standard not only in raster graphics editing, but in digital art as a whole.

Photoshop can edit and compose raster images in multiple layers and supports masks, compositing several color models including RGB, CMYK, CIELAB, spotcolor, and duotone. Photoshop uses its own PSD and PSB file formats to support these features. In addition to raster graphics, Photoshop has limited abilities to edit or render text and vector graphics (especially through clipping path for the latter), as well as 3D graphics and video. Its feature set can be expanded by plug-ins; programs developed and distributed independently of Photoshop that run inside it and offer new or enhanced features.

#### **4.1.2 Implementation Details of Modules (Description of procedures)**

##### **a) Movement Controller**

This controller is used to move all the elements in the game. Every surface walkable comprises of a navbase map which is used to detect collision as well as detect the location and position of the character. Whenever user inputs any movement keys( w,a,s,d, space) movement controller is responsible to handle the movement.

##### **b) Camera Controller**

This module is responsible for all of the camera movements in the game. There can be multiple camera in game to show various parts of the game window. This controller uses mouse events and other in game events to properly synchronize the camera movement in the game.

##### **c) Character State**

This module is responsible to handle all the characters in the game. Character features such as Health, Damage, Movement Speed, Jump height, all are managed by this module. Character State determines the state of the characters in the current frame in the game.

## 4.2 Testing

### 4.2.1 Test Cases for Unit Testing

**Table 1 : Unit Testing of Wrong Turn(A 3D Horror Game)**

SN	Function	Input	Expected Output	Result	Status
1	Move Character	W,A,S,D	Character movement	Character moved successfully	True
2	Pause	ESC	Pause screen	Game paused successfully	True
3	Resume	ESC	Resume Game	Game Resumed Successfully	True
4	Move Camera	Mouse input	Camera movement	Camera moved Successfully	True
5	Restart	Button	Restart game	Game Restarted Successfully	True
6.	Collect CD	E	Collect CD	CD collected Successfully	True
7	Spawn new CD		Spawn a new CD	CD spawned Successfully	True
8	Spawn Enemy		Spawn Enemy	Enemy Spawned Successfully	True
9	Play CD	F	Play CD	Played Successfully	True
10	Save Score	button	Save Score	Score Saved	True

#### 4.2.2 Test Cases For System Testing

**Table 2 : System Testing of Wrong Turn (A 3D Horror Game)**

SN	Function	Input	Expected Outcome	Result	Status
1	Spawn at the home		Spawn	Spawned	True
2	Aggro Enemy		Aggro the enemy	Enemy Aggroed Successfully	True
3	Trigger movement Animations	W,A,S,D	Run animations	Animations triggered Successfully	True
4	Trigger Enemy Attack	WASD	Trigger Enemy Attack	Enemy Triggered Successfully	True
5	Take Damage onhit		Take damage	Health reduced	True
6	Die when health 0		Die	Died Successfully	True
7	Collect CD	E	Collect CD	CD Collected and counter updated	True
8	End Game		End	Game Ended Successfully	True
9	Toggle Lights	F	Toggle lights	Lights Toggled Successfully	True
10	Open Doors	F	Open Doors	Doors Successfully	True

## **CHAPTER 5**

### **CONCLUSION AND FUTURE RECOMMENDATION**

#### **5.1 Lesson learnt / Outcomes**

There have been several improvements in our programming language and writing skills as well as our time management skills while doing this project. I conclude that this project has helped me gain more knowledge about the topic that we are indulged ourselves into “Unity Framework”. A lot was learned about proper time management as the project had to be submitted before the deadline along with the documentation.

Due to time constraints, I could not add more facilities to it. Although it is expectedly good, some new features to this system could be added in the upcoming days to make it more user friendly and efficient.

#### **5.2 Conclusion**

This project is done as Efficient as possible. The description of the background and the context of the project was thoroughly researched by the author.

The purpose, scope, applicability, and requirement specifications of the system have been accurately explained. The author has included features and operations in detail including screen layouts and the limitations on which the project is being developed. Finally, the system is implemented and tested according to test cases. After the development of the system finally, it was tested and the views about results were exchanged. After testing, the limitations of the existing system were discussed.

In conclusion, tools like Photoshop, Blender and backend tools like C# were used in the development of my system. I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future

### **5.3 Future recommendations**

The applicability and usage of this project can be increased by enhancing the system based on the limitations. Entertainment is now a basic need. Every Home has their own personal Computer. There are many people who need entertainment. So, with the help of this game we can deliver a good service to customer who wants some entertainment into. This helps to uplift the mental wellbeing of people. New effective's modules can be also added from time to time.

In future, new game modes can be added to the system where player can interact with even more objects. More interesting events can be added to the game with improved animation and graphics.

## REFERENCES

- [1] C. and J. , " Outlast review: Fraught in the dark," *Joystiq*, no. 5, pp. 10-25, 2013.  
[Accessed on: August 10, 2023]
- [2] C. Marshall, "A janky ghost game is blowing up on Twitch," *Phasmophobia*, pp. 10-25, 2020. [Accessed on: August 13, 2023]
- [3] A. Rollings and A. Ernest , "Fundamental of Game design," *Prentice Hall*, vol. II, pp. 2-12, 2006. [Accessed on: August 13, 2023]
- [4] Morton and Lauren , "Phasmophobia has certainly possessed a lot of players," *Rock, Paper, Shotgun*, vol. II, 2020. [Accessed on: August 14, 2023]
- [5] Apperley and H. Thomas , "Genre and game studies," *Simulation & Gaming*., p. 6–23, 2013. [Accessed on: August 14, 2023]
- [6] M. and C. , "Top 25 Facebook Games for November 2, 2009," *Inside Social Games*, no. 8, pp. 20-35, 2009. [Accessed on: August 16, 2023]
- [7] B. and N. , "Zynga Gets Serious," *Digits. Wall Street Journal*, vol. IV, pp. 30-40, 2010. [Accessed on: August 16, 2023]



## APPENDICES



