



Republic of the Philippines  
**RIZAL TECHNOLOGICAL UNIVERSITY**  
Cities of Mandaluyong and Pasig  
**COLLEGE OF ENGINEERING**

---



**MAS (Manggahan Application Services): Mobile-based Document Application for  
Barangay Manggahan, Pasig City**

**A Capstone Project & Research II**

**Presented to the Faculty of the Engineering, Architecture and Technology**

**RIZAL TECHNOLOGICAL UNIVERSITY**

Bgry. Malamig, Boni Avenue

Mandaluyong City

In Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Information Technology

**Department of Information Technology**

By

**BINUYA, RYAN CHARLES P.**

**BUENA, PAUL JUSTIN C.**

**GABRIEL, MAY LALYN C.**

**SARMIENTO, JERMAINE RICHARD V.**

May 2024



## APPROVAL SHEET

This Capstone Project entitled, ***“BLADESCAPE: SINGLE-PLAYER ACTION GAME”***, prepared and submitted by RYAN CHARLES P. BINUYA, PAUL JUSTIN C. BUENA, MAY LALYN C. GABRIEL, JERMAINE RICHARD V. SARMIENTO in partial fulfillment of the course requirements for the degree of BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY, has been examined and recommended for acceptance and approval for Oral Examination.

**Dr. Jaevier A. Villanueva**

Adviser

Approved by the Committee of Oral Examination on May 7, 2023 with a grade of \_\_\_\_.

**Dr. Lea S. Nisperos**

Chairman

**Mr. Jayson Daluyon, DITc**

Technical/Expert

**Prof. Reynaldo G. Alvez**

Local

Accepted as partial fulfillment of the course requirements for the degree in Bachelor of Science in Information Technology.

**DR. CORLETO R. BRAVO**

Dean, College of Engineering

Date: \_\_\_\_\_



## ACKNOWLEDGMENT

The proponents would like to extend their heartfelt gratitude and appreciation to all people who extended their assistance, encouragement, and support for the accomplishment of this project.

To the **Almighty God**, for being the source of courage, hope, and strength, for bringing wonderful people who contributed to the project, and for never-ending blessings and grace showered into their lives.

To **Engr. Greta M. Rosario**, their respective Capstone Professor, for her time, expertise in imparting her knowledge and encouragement to start our project – allowing the proponents to develop their system.

To **Dr. Jaevier A. Villanueva**, their eminent Capstone Adviser, for his time, and intellectual support in the project.

To **Mr. Jayson Daluyon**, the IT Expert - a member of the panel of examiners, for his valuable insights, suggestions, and sharing his viewpoint to make our system more efficient.

To **Dr. Lea S. Nisperos**, the chairperson, for her constructive suggestions, which contributed a lot in the improvement of the project.

To **Prof. Reynaldo G. Alvez**, a member of the panel examiners, for his valuable inputs and suggestions which contributed to the project's improvement.



## ABSTRACT

Title : **Bladescape: Single-Player Action Game**

Proponents : **Jermaine Richard V. Sarmiento**  
**May Lalyn C. Gabriel**  
**Paul Justin C. Buena**  
**Ryan Charles P. Binuya**

Capstone Adviser : **Dr. Jaevier A. Villanueva**

Capstone Professor : **Engr. Greta M. Rosario**

School Year : **2022-2023**

Course : **Bachelor of Science in Information Technology**

---

Technology has significantly impacted our daily lives allowing us to access resources and information more efficiently. Online processing has reduced the time spent on obtaining documents, allowing for greater efficiency and productivity. It also offers convenience and accessibility, eliminating the need for long queues and enhancing data security. Barangay Manggahan often faces difficulties in processing documents like barangay certificates and indigency. The traditional method of processing documents is not suitable for the barangay. This issue is particularly problematic for Pasig City programs, such as Pasig City Scholarships. Barangay Manggahan lacks a website or application to keep residents informed about events and resolve complaints. Researchers propose an app-based file processing document for Barangay Manggahan, which would benefit both barangay employees and residents by allowing them to process their documents anytime and anywhere.



The project would be created with a network for the app, managed by an admin, and ensuring the privacy of information. The mobile application services, in collaboration with the barangay, would also allow for the management of concerns through a platform for people to submit their legitimate concerns. The study aims to develop and implement a mobile application to provide swift and alternative ways of processing documents, validate documents more effectively, and expedite processing and requesting documents for barangay staff and residents. This will benefit the residents of Barangay Manggahan, barangay officials, and future researchers by providing better solutions for queueing, reducing delays, and tracking concerns and complaints. The mobile application will also include features for users, such as local news updates and a concerns page for easy complaint submission.

*Keywords: Mobile application, Application services*



## TABLE OF CONTENTS

<b>TITLE PAGE .....</b>	<b>i</b>
<b>APPROVAL SHEET .....</b>	<b>ii</b>
<b>ACKNOWLEDGMENT .....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>TABLE OF CONTENTS.....</b>	<b>vi</b>
<b>LIST OF TABLES .....</b>	<b>ix</b>
<b>LIST OF FIGURES .....</b>	<b>x</b>
<b>Chapter I INTRODUCTION .....</b>	<b>1</b>
Project Context.....	1
Purpose and Description of the Project .....	3
Objectives of the Project .....	3
Significance of the Study.....	4
Scopes and Limitations of the Project.....	5
Scope .....	5
Limitations .....	7
Operational Definition of Terms .....	8
<b>Chapter II REVIEW OF RELATED LITERATURE.....</b>	<b>10</b>
Related Literature .....	10
Related Studies.....	15
Synthesis.....	16



<b>Chapter III TECHNICAL BACKGROUND.....</b>	<b>18</b>
Technicality of the Project .....	18
Details of Technologies Used.....	19
How the project will work.....	22
<b>Chapter IV METHODOLOGY .....</b>	<b>23</b>
Environment (Locale and Population of the Study).....	23
Population Frame and Sampling Scheme .....	23
Description of the Respondents.....	24
Statistical Treatment of Data.....	24
Requirements Specifications .....	26
Operational Feasibility .....	26
Technical Feasibility .....	28
Schedule Feasibility .....	29
Economic Feasibility .....	30
Requirements Modeling .....	31
Wireframe .....	33
Requirements Documentation .....	36
Data and Process Modelling.....	36
Object Modelling.....	41
Risk Assessment / Analysis .....	45
Design of Software Systems Products and/or Process .....	46
Output and User-Interface Design.....	46



Data Design .....	56
System Architecture .....	63
<b>DEVELOPMENT AND TESTING .....</b>	<b>65</b>
Development .....	65
Testing.....	69
Description of the Prototype.....	76
Program Properties .....	77
Function Properties .....	77
Implementation Plan .....	78
<b>Chapter V CONCLUSION AND RECOMMENDATIONS.....</b>	<b>79</b>
Presentation, Analysis, and Interpretation of Data.....	79
Summary of Findings .....	85
User Experience Results.....	85
Conclusion.....	87
Recommendations .....	89
<b>BIBLIOGRAPHY .....</b>	<b>91</b>
<b>APPENDICES</b>	





## LIST OF TABLES

Table 1	Population Frame of the Respondents.....	23
Table 2	Four-Level Likert Scale Used to Evaluate Bladescape .....	25
Table 3	Technical Requirements for Computer Game.....	28
Table 4	Gantt Chart .....	29
Table 5	Cost Summary.....	30
Table 6	Risk Matrix in Bladescape's Game Development .....	45
Table 7	Saved Progress Data.....	57
Table 8	Third Person Controller Class Variables.....	59
Table 8.1	Third Person Controller Class Variables (Continuation).....	60
Table 8.2	Third Person Controller Class Variables (continuation).....	61
Table 9	Player Health Attributes .....	62
Table 10	Sound Variable Class .....	62
Table 11	Software Specification.....	65
Table 12	Implementation Plan.....	78
Table 13	WM and VI of the Functionality Evaluation.....	79
Table 13.1	WM and VI of the Reliability Evaluation .....	80
Table 13.2	WM and VI of the Usability Evaluation .....	81
Table 13.3	WM and VI of the Efficiency Evaluation.....	82
Table 13.4	WM and VI of the Portability Evaluation .....	83
Table 13.5	WM and VI of the Maintainability Evaluation.....	84



## LIST OF FIGURES

Figure 1	Fishbone Diagram.....	27
Figure 2	Input-Output-Process System .....	31
Figure 3	Wireframe .....	33
Figure 4	Game Environment Concept Map (Tutorial – Stage 2).....	34
Figure 4.1	Game Environment Concept Map (Stage 3 – Stage 4).....	35
Figure 5	Context Diagram.....	36
Figure 5.1	Player-Enemy Context Diagram.....	37
Figure 5.2	DFD Diagram Level 0 .....	37
Figure 5.3	DFD Diagram Level 1 .....	38
Figure 5.4	DFD Diagram Level 2 .....	38
Figure 6	Bladescape's Program Flowchart .....	39
Figure 7	Use Case Diagram .....	41
Figure 8	Sequence Diagram .....	42
Figure 9	Activity Diagram .....	43
Figure 10	Class Diagram.....	44
Figure 11	Main Menu (Initial Design).....	46
Figure 11.1	Settings (Initial Design).....	47
Figure 11.2	Paused Menu (Initial Design) .....	47
Figure 11.3	Settings - Paused State (Initial Design) .....	48
Figure 11.4	In-game UI (Initial Design) .....	48



Figure 11.5	Bladescape Game Map (Top View).....	49
Figure 12	Tutorial Map.....	50
Figure 13	Stage 1 .....	50
Figure 13.1	Stage 2 Upper and Lower Level .....	51
Figure 13.2	Stage 3.....	51
Figure 14	Boss Fight Map.....	52
Figure 14.1	Main Menu (Actual Output) .....	52
Figure 14.2	Settings (Actual Output) .....	53
Figure 14.3	In-game UI (Actual Output).....	53
Figure 14.4	Paused Menu (Actual Output).....	54
Figure 14.5	Settings in Paused Menu (Actual Output).....	54
Figure 14.6	In-game UI (Initial Design).....	55
Figure 15	Entity Relationship Diagram .....	56
Figure 16	Game Architecture of Bladescape .....	63
Figure 17	Deployment Diagram.....	68
Figure 18	Unit Testing Tutorial Stage .....	70
Figure 18.1	Unit Testing Stage 1 .....	71
Figure 18.2	Unit Testing Stage 2 .....	71
Figure 18.3	Unit Testing Stage 3 .....	72
Figure 18.4	Unit Testing Stage 4 .....	72
Figure 18.5	Unit Testing Boss Stage.....	73
Figure 19	Performance Testing Using MSI Afterburner.....	74



## **Chapter I**

### **INTRODUCTION**

#### **Project Context**

Action games require the player to strategically control their character while using weapons to win. It has complex controls for fighting, throwing, running, jumping, and other actions that an action game may require depending on the gameplay. Action video games, according to Nanou (2021), start out with peaceful quests and challenges before switching to a mode that challenges the player's hand-eye coordination, ability to complete a level under pressure, and reaction time.

Bladescape is a single-player, three-dimensional action game based on the 2019 3D Metro game series developed by 4A Games and published by Deep Silver. Metro is a survival horror first-person shooter action game that can be played on a variety of platforms, including Xbox, PlayStation, and Microsoft Windows.

Bladescape is a third-person shooter action game, which means that the player or main character must view the levels from a third-person vantage point. To win the game, the player must participate in fight in certain complex stages. The game's protagonist, who is also aware of his identity, awakens in an odd laboratory and must escape in order to unravel the mysteries behind his condition. The player's choice of weapon must be used to



combat enemies, elite rouges, and the boss that the main character will confront within the lab.

Today's popular action games, especially those that fall under the action category, are mostly played online as character-based tactical games, like Valorant, Call of Duty, Counterstrike, etc. However, Bladescape is a locally stored game that aims to implement a completely free-to-play gameplay experience which will allow gamers to download and play Bladescape without worrying about their financial stability in a pay-to-win era. If the player wishes to save the game, Bladescape has a saving feature that allows users automatically save their progress data.

Artificial intelligence has developed and enabled modern technology to adapt and innovate, making the science of the future a reality. Baker and Smith (2019) defined artificial intelligence as machines that carry out cognitive functions normally associated with human minds, including learning and problem-solving. Bladescape will employ AI-controlled NPCs with FSM algorithm to provide a quality action game to action game enthusiasts.



## **Purpose and Description of the Project**

Bladescape is made to elevate traditional action gameplay by engaging the players with a highly interactive narrative and NPCs with the help of AI (artificial intelligence). The enemies and bosses in the game will not be limited to the predictable strategic patterns that old action games used to challenge the players to beat the game.

Bladescape will have a certain number of main stages with enemies patrolling around the game map. Every stage has its own checkpoint to track the player's progress in the game. The story will unveil the mystery of the story once the player finishes all the stages in the story mode.

## **Objectives of the Project**

The primary objective of this study is to develop a single player action game with AI-controlled non-playable characters (NPCs) that will be played by gamers for a challenging and engaging gameplay.

Specifically, this study focuses on the following:

1. To build a game that incorporates artificial intelligence as an element of the game with an engaging narrative and challenging obstacles.
2. To develop a game with save game feature that allows the player to save their progress.
3. To deploy the game in a game distribution platform to make it available for users.



### **Significance of the Study**

Significant research is being conducted to discuss the game with students and fellow IT students. Future researchers can use this study to build their own game, as well as to innovate, experiment, and expand the scope of their research.

### **To the Researchers**

The researchers will acquire experience and knowledge, especially during the whole development period of the study. This study may be a big help to the researchers for their future, for it will be a part of their own portfolio when it is completed and published.

### **To the Institution**

This study will aid in the growth and motivation of the faculty members and students in the institution to carry out further research relating to game development. By educating the students with skills, knowledge, and even experience in game development field, this may encourage game development studies among the students in the institution.

### **To the future IT students and researchers**

This study will be contributory to future researchers to create and even undertake an extensive dissertation through this study by using it as a reference to learn how a game is built and to be aware of the resources that are required to build a successful game.

### **To Non-Computer Major Students**

This study will provide entertainment and information about action games and their creation and development.



## **Scopes and Limitations of the Project**

This study focuses on developing an action game with the use of artificial intelligence. In particular, the integration of artificial intelligence will focus on the non-playable characters to provide an advanced and fun gaming experience for the players. Further discussion of the scope and limitations is provided below.

### **Scope**

1. The game focuses on the narrative with the main character fighting enemies in order to find a way out of the strange laboratory. As the game progresses, the difficulty of interactions with the enemy NPCs will increase.
2. The game involves artificial intelligence-controlled enemies to present a challenge for players to finish the game.
3. The game consist of menus such as the Main Menu, Paused Menu and Game Over Menu.
4. In the Main Menu and Pause Menu, there is a Settings page that allows the user to configure the volume and sensitivity according to their gaming preferences.
5. The game implements a third person shooter perspective of the player controller for wider view of action gameplay.
6. The game application features an automatic save game feature that allows the player to save the game's progress whenever a level checkpoint is triggered.





7. The stages in the game consists of three basic stages with elite bosses, and a boss fight.
8. The upgrade element of the game consist of buffs and weapons that can be equipped by the player to match with the difficulty of the enemies in the map.
9. The weapon wheel found in the in-game screen features weapons that can be used one at a time by triggering keys intended for the function. The weapons consist of a knife, the sword, and a gun.
10. The buffs in the game lets the player obtain a certain ability in a small duration of time. The buffs are the attack speed and damage boost.
11. The game application presents a 360-degree view of the gaming environment and provide a high-definition quality of the gaming scenes as well as quality graphic animations with 3D elements to give an engaging experience in gaming.
12. Upon executing the game, a disclaimer will be present at the beginning to remind players that the game consists of graphic content and players under the age of 13 should play it with parent's supervision.
13. The game application is deployed on a digital platform that provides game distribution service to make the game discoverable and accessible for gamers.



## **Limitations**

1. Bladescape is a game under the horror action genre that may only be suitable for users or viewers under the age of 13 and above. With this, the game's disclaimer will be shown at the start of the game.
2. Bladescape is only being developed to function on a computer or a laptop with Windows 10 installed and is only limited to keyboard and mouse support for user-input.
3. The game does not support multiplayer.
4. Only non-playable characters are supported by artificial intelligence.
5. Some AI NPCs in the game levels are programmed to attack the player only when a predetermined distance is reached. This is due to the limitation of space in some areas of the game maps. Hence, some AI NPCs cannot patrol when the map's space is too crowded.
6. To prevent the levels from being too challenging or impossible for the player to complete, the AI NPCs who may patrol are only placed in levels with large open areas. By doing this, the game's difficulty and the map's crowd spacing are both balanced.
7. The game's saving feature is offline based.
8. The game is limited to its Beta Version.



## **Operational Definition of Terms**

**Action Game** - Action games are a subgenre of video games that place a strong emphasis on physical difficulties like quick reflexes and hand-eye coordination.

**Artificial intelligence** – or known as AI, is the emulation of aspects of human intellect by machines, primarily computer systems.

**Computer Game** – a game application that is being run on a computer and enables the player to interact with objects that are shown on a screen.

**C#** - often known as C-Sharp, aims to combine the processing power of C++ with the programming simplicity of Visual Basic.

**Finite State Machine (FSM)** - an algorithm that generalizes every game-playable event and then executes a script for a particular response to that particular event.

**HUD** – or Heads-Up Display, which is all what is displayed on the screen while the player is in-game.

**Integrated Development Environment (IDE)** - An integrated development environment (IDE) is a tool for building applications that integrates well-known developer tools.

**Main Character** – the character who is controlled by the player as the protagonist in the game.

**Non-Playable Characters (NPCs)** - All characters in a game who cannot be controlled by a player.



**Player** - A player is a person who uses a platform for entertainment, such as a game, a social network, or another similar system.

**Sound Effects (SFX)** – is the special audio effects that are utilized in the game to affect the mood and reactions of the users.

**Stages** – or also referred to as level, are any areas the player has access to while completing a task. It is a point in the game of an ongoing process.

**Third Person Controller (TPC)** – is responsible for controlling the player's movement by handling inputs from a keyboard and mouse and a gamepad.

**Third-person Shooter (TPS)** - a kind of computer game where the player shoots targets while positioning the viewer's perspective behind or around the player's character.

**Three-dimensional (3D)** - 3D means three-dimensional, i.e., something that has width, height, and depth (length).



## **Chapter II**

### **REVIEW OF RELATED LITERATURE**

Numerous studies on the creation of games and the application of artificial intelligence to various technological fields have been conducted. The research supporting the inclusion of artificial intelligence in the creation of action games is summarized in this chapter.

#### **Related Literature**

Games have been part of the people's source of entertainment since the 1950s. To define a game, there are a lot of factors to consider including its very nature and effects to its users. Since the 1930s, there have been more than 60 definitions of the word "game". According to Suits (2022), playing a game involves engaging in activity intended to bring about a particular state of affairs using only methods allowed by a set of rules, which are more restricted in scope than they would be in the absence of the rules, and where the only justification for accepting such restrictions is to enable such activity. (Suits, 2022)

Games have been evolving since then starting from board games, from more than 5000 years ago, to today's computer games. Arjoranta (2019) asserts that there are philosophical justifications for why it's crucial to define "game" and why a new definition is required. This is due to the fact that games are a constantly evolving cultural



phenomenon. People's perspectives on games continue to change as the culture around them does, even if the ideas themselves do not. (Arjoranta, 2019)

Bladescape is a game with a narrative and set of rules to motivate its players to progress as they follow the game's story. Its mechanics is to fight the enemies within the environment and progress with the game's narrative. It is designed to be played in a computer to allow the users to interact with things displayed on a screen using a set of controls from the keyboard and perspective movement from the mouse. With this, the game becomes engaging, fun, and challenging for the players.

The researchers chose computer as Bladescape's platform because it offers a variety of keys and controls, enhancing the player's hand-eye coordination and response time. According to Study.com (2023), computer games are a form of interaction with games that can be played anywhere there is a computer. In contrast, the term "video game" refers to any interactive game that may be played on an electronic device. Video games include those played on different platforms including video game consoles and mobile phones as well as those played on computers. (Study.com, 2023)

Given that the rules and mechanics, and the game's platform for gameplay is already identified, Bladescape can be now categorized under the action game genre. According to Nanou (2021), "action games" is a game genre that challenges the player physically as the name suggest that the essence of action game should be more active and engaging. The significance of the player's sensorimotor abilities (which include both hand-



eye coordination and response time) in carrying out the numerous actions required to advance through the game's difficulties therefore appears to be the distinguishing characteristic of "action" games. (Nanou, 2021)

Bladescape belongs not only to the action games genre but also can be classified under the action as a mode of gameplay. Nanou (2021) wrote that action as a mode of gameplay are experienced and separated into three things directly: physical involvement; movement; and character's death in the gameplay. (Nanou, 2021)

To make the game more challenging, the idea of playing against bots has caught the attention of the researchers. Playing against bots has been a fundamental element of gaming since the earliest days. Togelius (2022) wrote that playing against bots simply couldn't provide the same challenge or excitement as competitive online play, so the introduction of multiplayer games in the late 1990s largely made game-playing bots an afterthought in many genres. In this sense, multiplayer games came into being because it seemed cliché for players to compete against a computer that is preprogrammed with known patterns and strategies. (Togelius, 2022)

With this regard, the researcher came up with the idea to build Bladescape with AI-supported NPCs who will be the player's opponent in the game. With the help of AI, action game enthusiasts can anticipate a more sophisticated and improved gaming experience with the integration of artificial intelligence.



According to Togelius (2022), the non-playable characters (NPCs) that are common in adventure games can become much more dynamic and interesting with the help of AI. Using AI could cause NPCs to respond in more unexpected ways rather than just following a pre-programmed cue to give a predictable response. Additionally, it could greatly simplify the creation of NPCs, whose complexity currently restricts the variety and level of characters that players can interact with in video games. Additionally, it's a common complaint from players that NPCs that are supposed to help them frequently end up obstructing gameplay or thwarting their progress. With improved player modeling, AI agents can reasonably anticipate what the player will do next and make plans in accordance with that information to assist the player in achieving their objectives. (Togelius, 2022)

In this study, the researchers utilized an algorithm such as Finite State Machine (FSM) to make the artificial intelligence in the game feasible. According to Buttice (2022), artificial intelligence in games as a way to enhance players' gaming experience by engaging the players to AI's unpredictable moves. This claim aligns with the researchers' aim – which is to enhance the player's sensorimotor abilities, creating a competitive and challenging gameplay scenario. Developers use algorithms such as Finite State Machine (FSM) to create a challenging game for the players. With this algorithm, AI NPCs will appear more intelligent compared to those who follow preprogrammed patterns in games, particularly those of traditional genres. (Buttice, 2022)

The Finite State Machine method, as described by Buttice (2022), is an abstract model of computing that may exist in one state at a time (current state), selected from a





limited range of values. If the idea is applied in games, the algorithm generalizes every game-playable event, and then executes a script for a particular response to that particular event. In this algorithm, the NPC is capable of patrolling, searching, engaging, and covering, depending on the player's movement in the environment and the NPC's line of sight. (Buttice, 2022)

Aside from employing AI NPCs in the game using FSM, in the development phase of this study, the researchers utilized Unity as Bladescape's game engine and editor.

According to Sinicki (2020), many developers use Unity as the platform and building block for their creations. As a game engine, it is able to provide many built-in features that make up games. It features collision detection, 3D rendering, and physics. Instead of building a game engine from scratch, Unity is able to offer everything game developers require. The Unity software includes practical tools and features that make it an effective IDE, including the ability to browse project repositories. Unity can also be used as an alternative programming editor, much like IDE programs like Microsoft's Visual Studio. (Sinicki, 2020)

Without having to deal with a lot of code, Unity enables users to complete many tasks in the software. The software gives you the flexibility to change almost everything and create more than the user can achieve. Unity uses C#, the most beginner-friendly programming language. It is extensively used in the sector and has many features in common with other well-known languages like C and Java. (Sinicki, 2020)



## **Related Studies**

To further support this project, several studies related to this capstone are being utilized as a support and reference in conducting and developing Bladescape, particularly in the artificial intelligence aspect of the project.

Lu (2020) presented a technique and system for controlling AI behavior in video games that enumerates many fixed-line behavior logic types through program coding and implements fundamental behavior logic control through table arrangement. The current mechanism for controlling game AI behavior logic is single-solidified, difficult to scale, and makes it challenging to implement specific game AI needs.

The process involved acquiring a game AI logic control tree, each node of which is used to characterize all of the game AI behavior logic; acquiring the target restriction condition of the game AI, which is a restriction condition imposed by the game environment on the game AI; and determining the logic control tree for the game AI based on the target restriction condition. (Lu, 2020)

In short, the AI behavior of the game are all determined using AI logic control trees and restriction conditions for determining the behavior of the logic.

Another study invented by Yang (2021) about a device that generates a boss battle fight type AI that improves the generation efficiency of the combat fighting AI game model. A combat fighting AI game model can be created using this application by first



obtaining a first combat fighting AI game model, then training the first pair of fighting AI game models using the multiple training samples to produce the second pair of fighting AI game models, where each trainable AI game character is based on the first combat fighting AI game model. (Yang, 2021)

### **Synthesis**

The reviewed literature primarily discusses the central idea of what constitutes an action game that incorporates artificial intelligence. Action games have long been a popular source of entertainment, particularly for players. AI creates the possibility for non-playable characters and sophisticated algorithms to advance for future in-depth research.

The related studies covered in this chapter helped the researchers by utilizing these studies as a reference in building the concept of artificial intelligence in games such as the invention of generating boss game type invented by Yang (2021) and the logic control tree method by Lu (2020) which inspired the proponents to further investigate about different algorithms that can be applied in AI NPCs.

Meanwhile, the literatures covered in this chapter helped the researchers put their newfound information and understanding of the subject matter to use. The researchers were able to categorize the genre of the game project as well as its mode of gameplay. With this, Bladescape was able to be distinguished as a computer game under the action genre. More importantly, the FSM algorithm made the project feasible as it introduces an algorithm that helped the researchers model the AI NPCs in the game. In the game, just like how Buttice



(2022) described FSM, the algorithm makes the AI respond or react depending on the player's movement.

In this study, the researchers used the platform to develop the game application starting from the game assets to the running and finalized build. Unity is a popular cross-platform game engine used to create games. According to Sinicki (2020), many developers use Unity as the platform and building block for their creations. As a game engine, it is able to provide many built-in features that make up games. (Sinicki, 2020)



## **Chapter III**

### **TECHNICAL BACKGROUND**

This chapter presents an overview of the project's technicalities and the details of the technologies used in building Bladescape.

#### **Technicality of the Project**

Bladescape is a single-player TPS action game that employs AI NPCs using the FSM algorithm to provide a challenging aspect without using predictable counters against the enemies in the gameplay.

After gathering all the resources and related studies to support the possibility of developing the game application, the researchers used the best method to build a game that will enhance the gaming experience of the players. Afterward, the game application is developed using C# as its programming language, and it is made available on the chosen distribution platform for users to download and play for free. The researchers were able to create a three-dimensional action game with intelligent NPCs having the capabilities of artificial intelligence (AI). Some of the terminology that the researchers have used include AI, third-person shooter (TPS), VS Code, Unity Editor, and C#. The researchers are using a few of the aforementioned terms to build the project.



## Details of Technologies Used

The researchers have collected all the technologies needed in developing the 3D action game Bladescape. All the details of the technologies used are discussed below:

### Enemy AI NPC

Using the Finite State Machine (FSM) algorithm, the enemy AI NPCs are capable of patrolling, searching, engaging, covering, and alerting other enemies. This method employs the FSM model, which includes tridimensional senses such as vision, hearing, and focus. These AI NPC enemies are mostly defined by their radius and view angles.

As discussed in Chapter II, The FSM algorithm has stages wherein the enemy will respond and counter depending on the player's movement. The following stages from the model applied in the Bladescape are Patrol, Cover, and the most important part, Combat. These are further discussed below:

1. The **patrol** state is a clear alert level in which the NPC monitors a path or a fixed location.
2. In the **engage** condition known as "**cover**," the NPC looks for cover to shield itself from direct fire. Like for instance, in the game environment, there is a wall nearby the NPC, and there is a navmesh path through the position, the NPC will run on that spot. In this state, the NPC also has the chance to advance and cover near the player's position to make the enemy more aggressive in the game, giving it a tightened level for the players.



3. Finally, the **combat** stage are behaviors that are brought on when the player and the enemy are engaged in close combat battle, causing the NPC to keep his attention on the target while moving and avoiding turning away from the player.

### **Game Assets**

The researchers gathered resources from websites, such as Unity Free Assets and Maximo, where users may obtain free gaming assets. This allowed Bladescape to be produced within the intended time range and without having to start from scratch. The researchers altered the assets using the Blender software and the Unity Editor to make the assets more cohesive and compatible with the visual style and aesthetics of the game.

The assets gathered are 3D models or prefabs of the pre-coded controllers, map, and props, materials, sound and visual effects, and animations.

### **Hardware**

#### **Computer**

A computer is an electrical machine that can be programmed to carry out specific activities like running general programs and processing and storing data. It can also serve as a medium for entertainment and communication (Rapaport, 2018). On the user's end, the gaming application will be executed utilizing an Intel Core I5 series laptop or PC with 8GB RAM and at least a 250GB+ SSD and the ability to run common programming applications.



## **Software**

A source code editor is a text editor designed for writing and editing source code by programmers that provides features such as compiling, interpreting, and debugging code (Podhradský, 2018). The researchers use the latest versions of the Unity Development Platform and Visual Studio 2022 Community to develop the game application.

To create and modify the 3D models and other game assets necessary to build Bladescape, the researchers used the Blender software and Unity Editor. These models include: the 3D models of the game environment (including the levels); the characters and NPCs; the materials, and prefabs (such as weapons, props, etc.).

## **Front-end and Back-end**

The front-end aspect of the game application will focus on three-dimensional assets. The developers will create 3D models for the game application using the most recent version of Blender and Unity Editor in order to achieve the three-dimensional aspect of the game.

The backend language of today's popular games is C#. The programming language is often used by game developers since it is an excellent option for creating desktop apps and game software and is well-known for its open-source community.

## **Peopleware**

The proponents are the developers, which consists of four individuals, and the players, who are potential users of the game application.





### **How the project will work**

The project is designed to have intelligent NPCs in the game to meet the action game standards of today's action games.

The project will work by installing the executable file of the game in the computer or downloading the beta version of Bladescape from the proponents' chosen game distribution platform. Upon making game progress, the game will save the player's progress locally.



## Chapter IV

### METHODOLOGY

This chapter presents an overview of Bladescape's technical specs, procedures, development, and testing. This chapter also discusses the research procedures, which include game documentation and study designs.

#### Environment (Locale and Population of the Study)

##### Population Frame and Sampling Scheme

The population for this study includes only gamers who own a computer or laptop device and IT practioners who have experience in examining game applications. A total of twenty (20) respondents comprised the total population frame of the study. These respondents reside from Philippines, Indonesia, and Malaysia as they are still qualified for the testing as long as they play computer games, and own a computer or laptop, or an IT practitioner: hence, purposive sampling was used.

**Table 1**

*Population Frame of the Respondents*

<b>Respondents</b>	<b>f</b>	<b>%</b>
Philippines	16	80%
Malaysia	2	10%
Indonesia	2	10%



## Description of the Respondents

To evaluate Bladescape through the game application in terms of functionality, reliability, usability, efficiency, portability, and maintainability, the researchers selected twenty (20) testers who own a computer or a laptop and IT practitioners as evaluators of the game “Bladescape”. The criteria were explained to them before they evaluated the game.

## Statistical Treatment of Data

Certain statistical treatments were applied to the data acquired in this investigation. The following are the statistical tools used in this study:

1. To determine the respondent’s evaluation of Bladescape in terms of Functionality, Reliability, Effectiveness, Usability, Portability, and Maintainability, the weighted mean was used.

Formula:  $\bar{WX} = \frac{\sum fx}{\sum f}$

Wherein:  $\bar{WX}$  = Weighted Arithmetic Mean

$\sum fx$  = sum of all the products of  $f$  and  $x$ , where the  $f$  is the frequency of each weight and  $x$  is the weight.

$\sum f$  = sum of all the respondents

The proponents utilized a four-level Likert scale, which is used for scaling responses in survey research, to identify the pertinent descriptive equivalent and



the responses of the respondents. The structure for typical four-level Likert items includes phrases like strongly disagree, disagree, and strongly agree, and agree.

**Table 2**

*Four-Level Likert Scale Used to Evaluate Bladescape*

Assigned Values	Range	Interpretation
4	3.00 - 4.00	Strongly agree
3	2.00 - 2.99	Agree
2	1.00 – 1.99	Disagree
1	0.99 – 1.00	Strongly Disagree



## **Requirements Specifications**

This section covers the feasibility studies which were established to improve the quality of the project respectively.

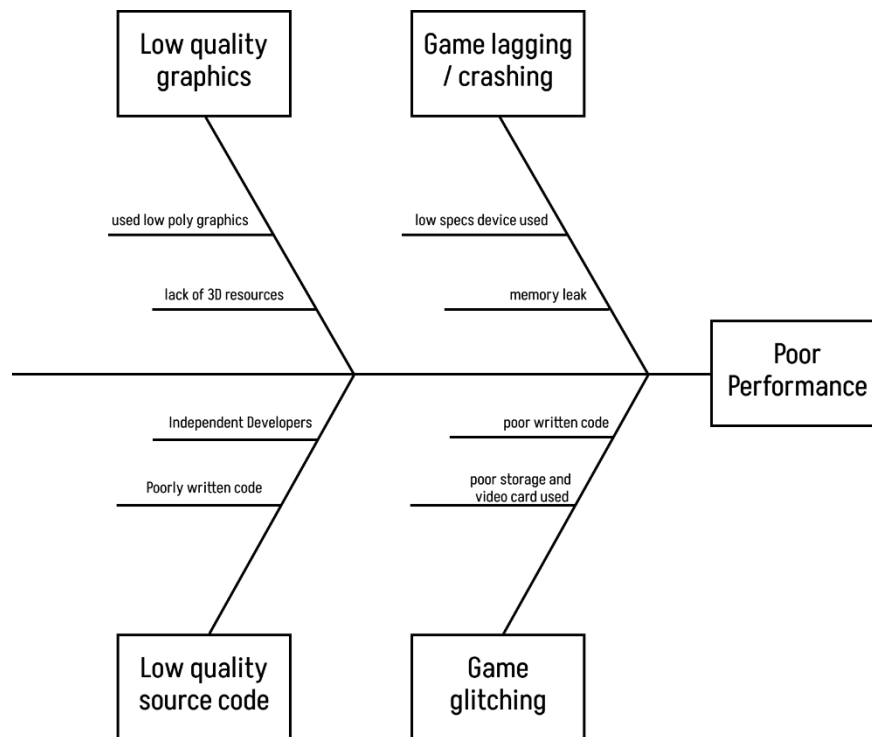
### **Operational Feasibility**

To make this project feasible, the proponents developed Bladescape as an offline-based game. As soon as the project is ready and polished, it is published on itch.io, an open marketplace for independent digital game creators. With this, Bladescape will be available for the users to download for free.

The website itch.io is a platform for independent developers to publish their games for free and allows its digital creators to make their games open for donations. With this, the digital creators have the opportunity to earn and make their games discoverable by a wide range of gamers in the community.

**Figure 1**

*Fishbone Diagram*



The various factors that could affect a problem are shown in the fishbone diagram. The gaming application's first use for development testing and test runs is highlighted as a possible source of issues. To avoid obstructing the core goal, it can assist the proponents in anticipating and resolving issues in advance.



## Technical Feasibility

### Compatibility Checking

**Table 3**

*Technical Requirements for Computer Game*

Game Application		Internet Connection
Hardware	Software	Broadband Cable or Fiber Optics Internet Connection
Intel Core Series, 4-core Modern CPU	Unity Editor, Unity 3D, Visual Studio 2022 Community Edition, Blender	
Memory: at least 8GB RAM and above		
Storage: at least 256GB SSD above	Operating System: Windows 10 above	
Monitor, Mouse, Keyboard		

The game application necessitates a minimum hardware specification of Intel Core Series or higher, with RAM of at least 8GB or higher. Storage should be at least 256GB SSD or higher. The researchers will use Unity Editor and Visual Studio 2022 Community Edition to design and program the game application.

### Relevance of the Technologies

Bladescape is a game application distributed by itch.io to provide entertainment to gamers with the enhanced action gameplay with the aid of AI NPCs. The hardware and software requirements as well as the connectivity to the internet, all have significant roles in running the game to maximize the gaming experience of the users.



## Schedule Feasibility

**Table 4**

*Gantt Chart*

Task name	Start date	End date	Assigned	18-Aug-22	1-Sep-22	15-Sep-22	29-Sep-22	13-Oct-22	27-Oct-22	10-Nov-22	24-Nov-22	8-Dec-22	22-Dec-22	5-Jan-23	19-Jan-23	2-Feb-23	16-Feb-23	2-Mar-23	16-Mar-23	30-Mar-23	13-Apr-23	27-Apr-23
<b>BLADESCAPE OVERALL PROGRESS</b>	18-Aug-22	7-May-23	Proponents																			
<b>Strategy, Planning, and Documentation</b>	18-Aug-22	19-Aug-22	Proponents																			
Brainstorming, Proposal of Topics	18-Aug-22	25-Aug-22	Proponents																			
Proposal Documents and Acceptance Letter	25-Aug-22	1-Sep-22	Proponents																			
Software Installation	25-Aug-22	2-Sep-22	Proponents																			
Preliminaries Chapter 1-3	25-Aug-22	3-Nov-22	Ryan/Lalyn																			
Oral Defense Documents	4-Nov-22	7-Nov-22	Proponents																			
<b>Software Design</b>	25-Aug-22	7-Nov-22	Proponents																			
Prototype	25-Aug-22	7-Nov-22	Proponents																			
Wireframe	25-Aug-22	15-Oct-22	Proponents																			
UI Design	25-Aug-22	15-Oct-22	Proponents																			
<b>Development</b>	18-Aug-22	15-Apr-23	Proponents																			
Polished Documents	28-Nov-22	17-Dec-22	Ryan/Lalyn																			
Revised Chapters 1-3	28-Nov-22	17-Dec-22	Ryan/Lalyn																			
Development	18-Aug-22	14-Apr-23	Proponents																			
<b>BLADESCAPE: SINGLE-PLAYER ACTION GAME</b>	18-Aug-22	5-May-23	Proponents																			
Module 1 (User Interface)	20-Aug-22	30-Apr-23	Jermaine																			
Module 2 (Steam Integration)	26-Apr-23	5-May-23	Proponents																			
Module 3 (AI Enemy NPCs)	20-Mar-23	30-Apr-23	Jermaine/Paul																			
Module 4 (Player Controller TPS)	18-Aug-22	30-Apr-23	Jermaine/Paul																			
Module 5 (Game Misc)	24-Feb-23	30-Apr-23	Jermaine/Paul																			
Documentation - Chapter IV	17-Feb-23	6-May-23	Ryan/Lalyn																			
<b>Project Testing and Development</b>	1-May-23	5-May-23	Proponents																			
Testing	1-May-23	2-May-23	Proponents																			
System Testing	1-May-23	2-May-23	Proponents																			
Integration Testing	1-May-23	2-May-23	Proponents																			
Alpha & Beta Testing	2-May-23	5-May-23	Proponents																			
Deployment	5-May-23	5-May-23	Proponents																			
Steam Deployment	5-May-23	16-Apr-23	Proponents																			
Documentation of deployment process	5-May-23	5-May-23	Ryan/Lalyn																			

*Note.* The project timeline of Bladescape development from August 2022 to May 2023





## Economic Feasibility

The cost and benefit analysis of the development in Bladescape are examined and considered in this section of the study.

## Cost and Benefit Analysis

All of the materials, including the publication costs and assets, have been gathered by the researchers to build Bladescape. These resources come in different price ranges and are taken into consideration whether it will be feasible for the researchers .

**Table 5**

### *Cost Summary*

Cost Summary			Project Budget	Project Actuals
Non-recurring	Game Assets	NPC Models	\$ 60	\$ 40
		Materials	\$ 10	\$ 5
		SFX & VFX	\$ 35	\$ 0
		Animations	\$ 50	\$ 0
		Maps Prefabs	\$ 50	\$ 0
		Controller	\$ 30	\$ 0
		Others	\$ 30	\$ 13.95
	Publishing Fee		\$ 100	\$ 0
Total		\$ 365.00	\$ 59	

*Note.* The resources bought in the Unity Asset Store are acquired in US dollars currency.

## Cost Recovery Scheme

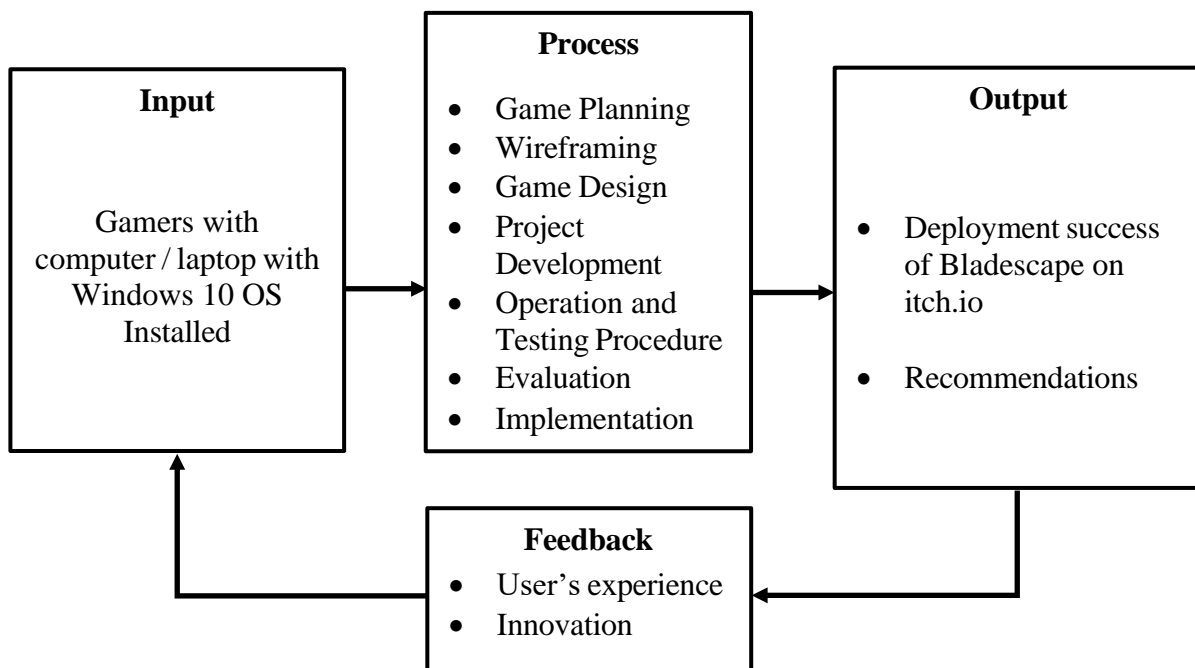
Since the researchers aim to publish a game that anyone may enjoy for free, Bladescape will be available to gamers and can be downloaded without fee. The researchers believe that players who do not monetize are also part of the community of the game.

## Requirements Modeling

This section describes and identifies the requirements of the game application Bladescape. These requirements are classified and illustrated below:

**Figure 2**

*Input-Output-Process System*



The figure above shows the input, process, and output of the project as it was being developed. The hardware, software, and programming languages needed to create the game are all included in the input diagram.

The actions that must be followed during the full research and implementation process are shown in the process diagram.

The output diagram shows the finished game. Additionally, the feedback explains how the player experienced about the game's functionality and quality.



### **Input**

Once Bladescape is successfully released on itch.io, the application's initial run will encourage the players to download the game on the mentioned marketplace and install the game on the computer. From then, Bladescape will be able to track the saved progress in the game upon making progress.

### **Output**

Bladescape will consist of four stages which includes the tutorial and the three levels that will make up the narrative of the game. The viewer's discretion will be advised at the beginning as it is expected that the game application will contain graphics that are unsuitable for children under the age of 13.

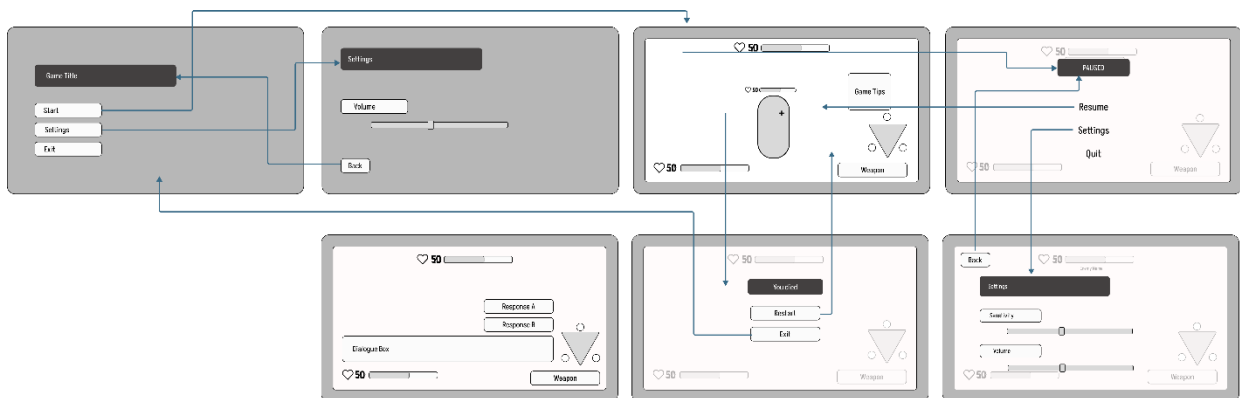
## Wireframe

The wireframe is a set of images designed to show the functional elements of an application or system. The proponents used Figma as a tool to design the projects wireframe.

The concept map of Bladescape is also shown below and will be based on its storyline so the main character will be able to reveal the story of the game.

**Figure 3**

*Wireframe*

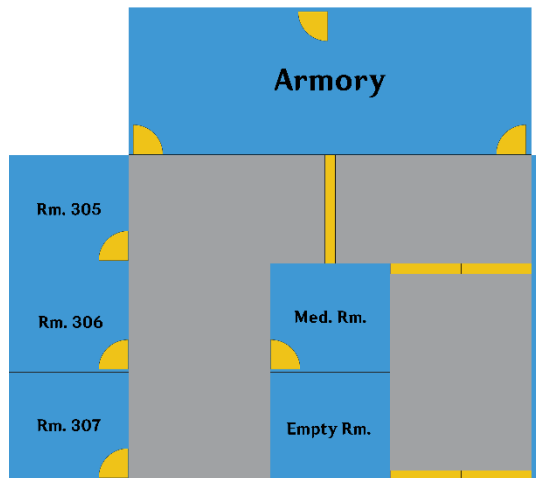




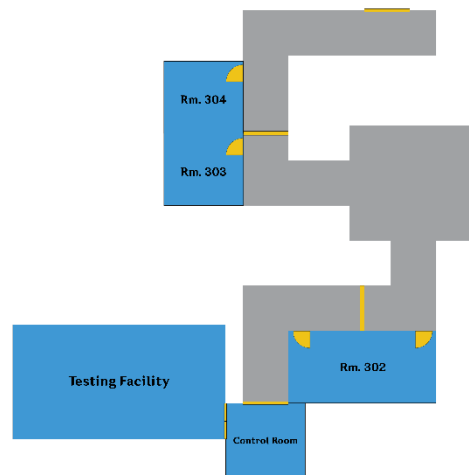
**Figure 4**

*Game Environment Concept Map (Tutorial – Stage 2)*

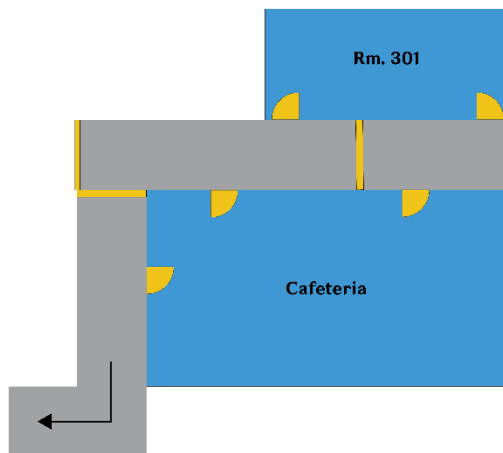
### Tutorial Concept Map



### Stage 1 Concept Map



### Stage 2 Concept Map (Upper Level)

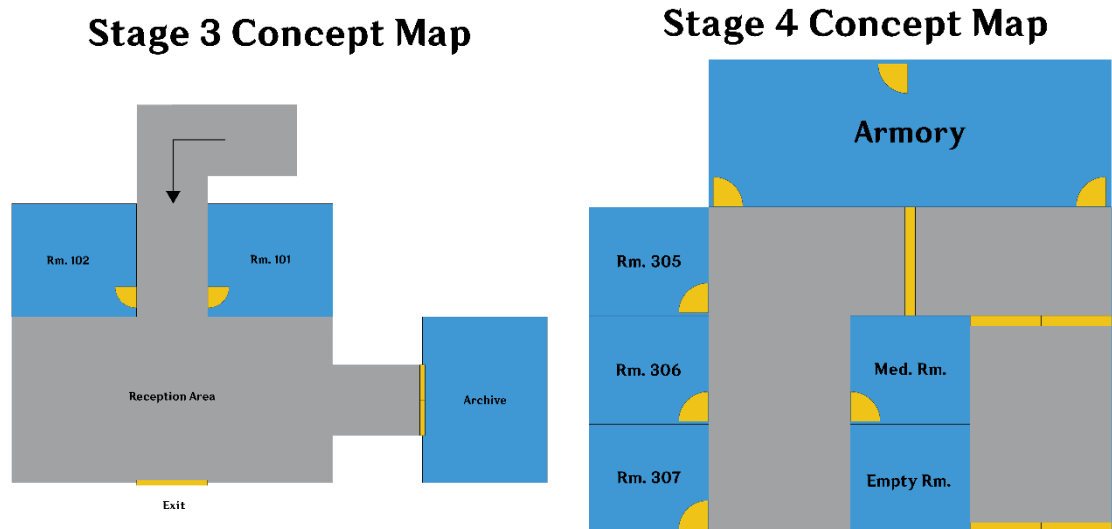


### Stage 2 Concept Map (Lower Level)



**Figure 4.1**

*Game Environment Concept Map (Stage 3 – Stage 4)*



*Note.* Concept maps from tutorial to the stage 4 of the game

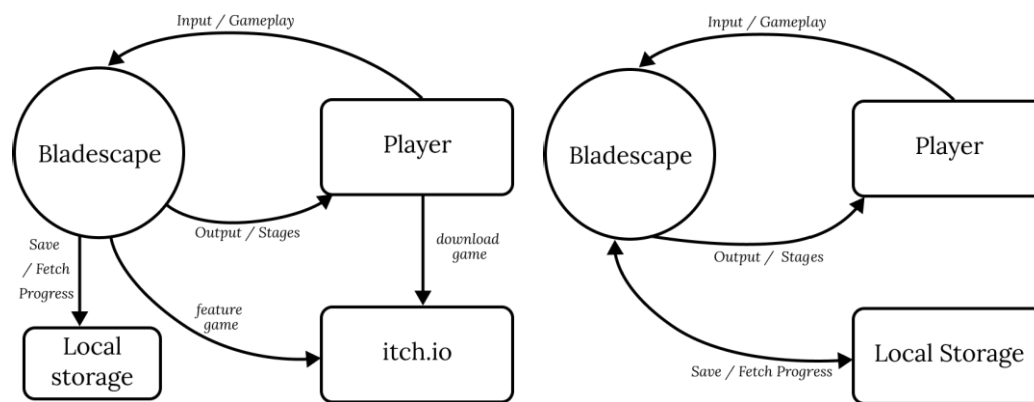
The researchers modeled the game map and plotted the game's scenarios using the concept maps as a basis. The tutorial level and stage 4 share the same map since they are both important parts of the game's plot, as well as the idea of the laboratory's exit being in the stage 3. The boss fight's concept map, which merely shows a large area to depict the boss' preferred battleground, is not illustrated in the concept maps. The actuals of the concept maps above are shown in the software design section of this chapter.

## Requirements Documentation

### Data and Process Modelling

**Figure 5**

#### *Context Diagram*

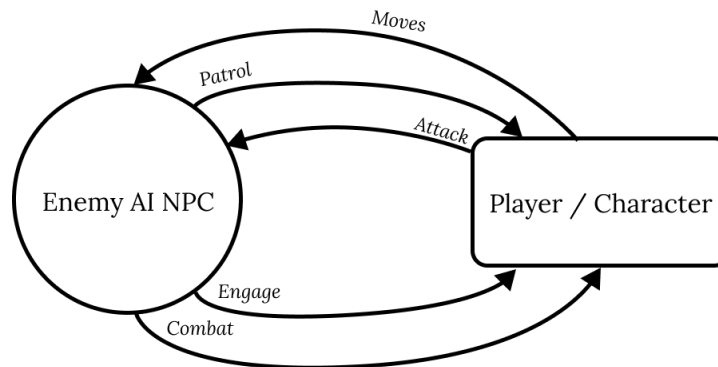


*Note.* Context diagram of the player and the game Bladescape.

The context diagram above illustrates how the user interacts with Bladescape, the game program, and the local storage and itch.io, which is the chosen distribution platform. The diagrams explain the two instances of Bladescape. The first instance is when the game is downloaded from itch.io. And the second instance illustrates its saving system with the user's local storage.

**Figure 5.1**

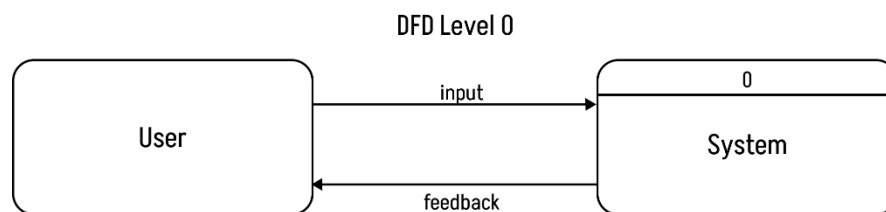
*Player-Enemy Context Diagram*



The context diagram above shows how the player interacts with the enemy AI NPC. This illustration is based on the FSM model. The AI NPC will patrol to engage in combat. Once an enemy NPC is aware of the player's location, the other NPCs around will be alerted to attack the player.

**Figure 5.2**

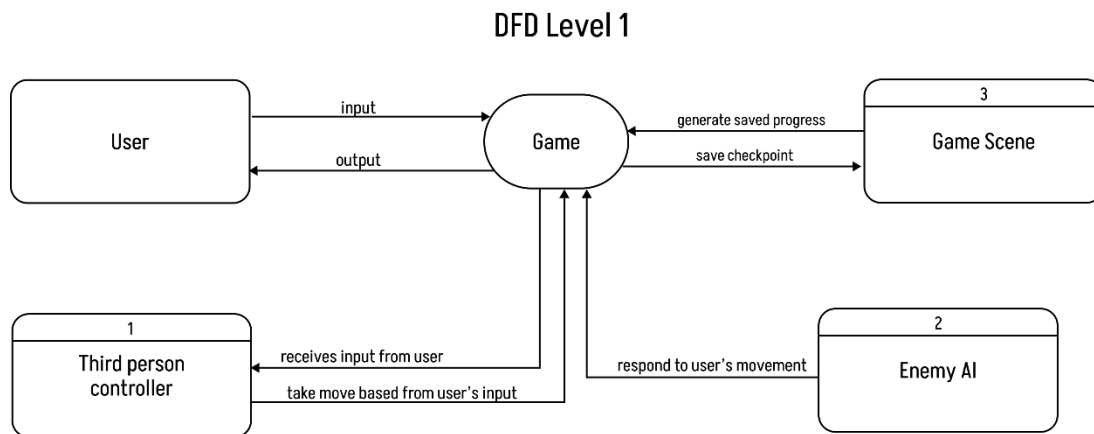
*DFD Diagram Level 0*





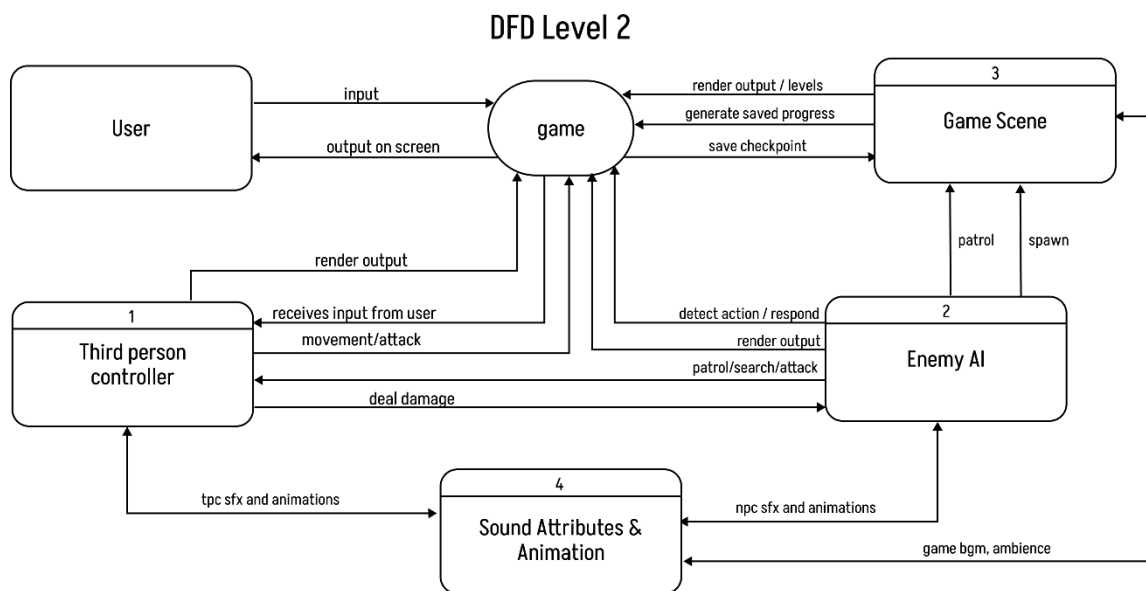
**Figure 5.3**

*DFD Diagram Level 1*



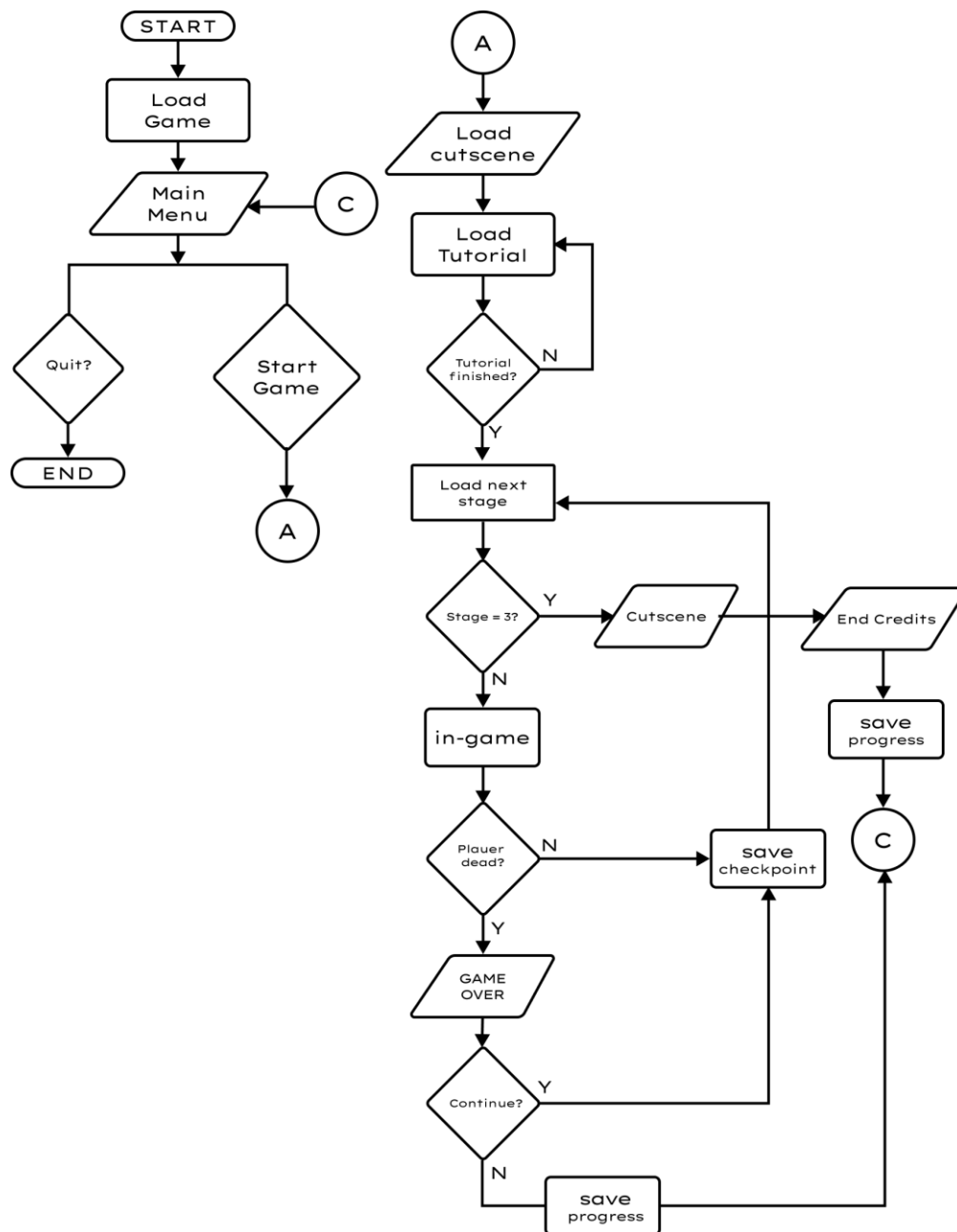
**Figure 5.4**

*DFD Diagram Level 2*



**Figure 6**

*Bladescape's Program Flowchart*



*Note.* This program flowchart illustrates the highlights of the game Bladescape



The shapes in the study's flowchart represent algorithms and processes. To give a more detailed explanation, the game application will be launched at the start of the process flow chart, displaying the main menu which consists of two main functions: Start Game, Settings, and the Exit.

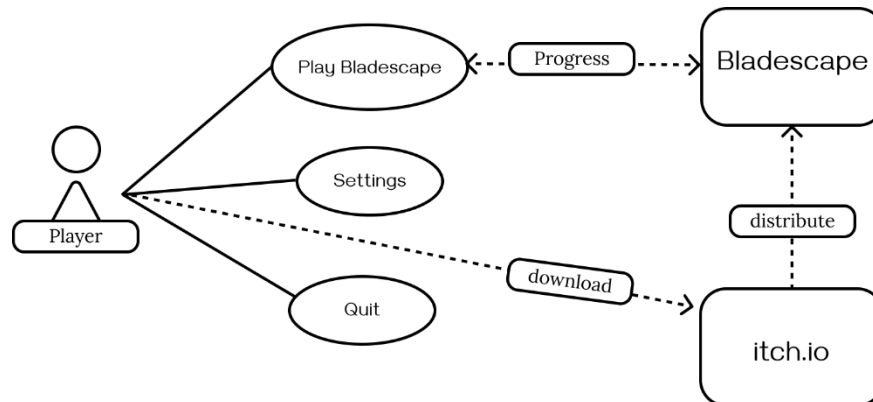
The game will begin with a tutorial in which a cutscene will flash on the screen introducing the story and the main character. Following the game scene, the tutorial will teach the player the game's basic controls and rules. Before proceeding to the first stage of the game, the player must first complete the first basic mission. If the player accomplishes the tutorial mission, they can proceed to the first stage of the game. If not, the player has to restart and try again.

In the game levels, the player must complete his task by eliminating every enemy ahead using a weapon equipped in the weapon wheel. If the player dies and fails to eliminate the enemies, the game will end, and the option will be prompted on the screen whether the player would like to try again or return to the main menu. The player's latest checkpoint triggered will be saved by then and can continue to the same checkpoint if the player resumes the game from the main menu. If the player successfully completes the current level, the player can enter the next level and will trigger the checkpoint. By then, the progress will be saved and will be able to continue on to the next stage.

## Object Modelling

**Figure 7**

*Use Case Diagram*

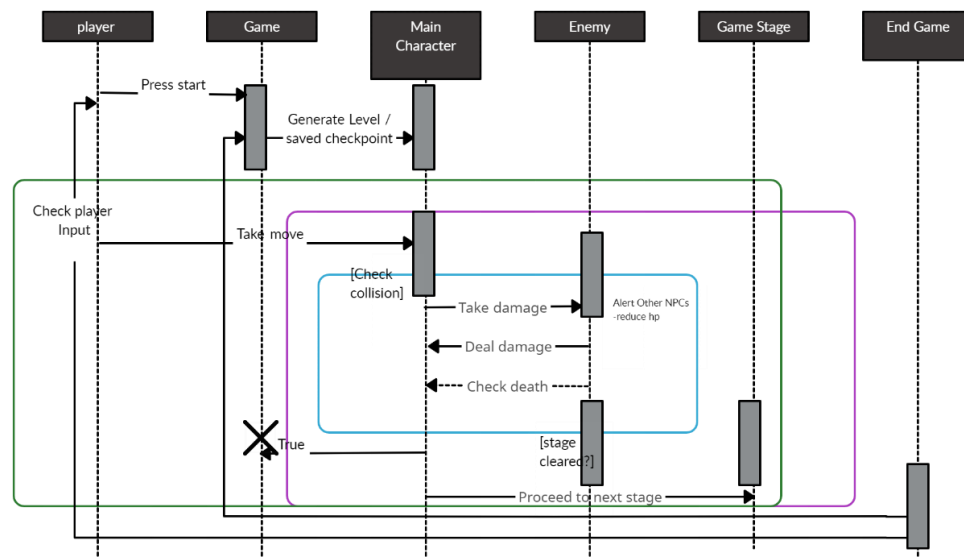


*Note.* Illustration of the interaction of the player to Bladescape's functionalities and the itch.io platform.

In the use case model shown above, the player interacts with Bladescape's fundamental features. This scenario happens when Bladescape is successfully deployed on itch.io and is acquired by the users using the platform. Additionally, the itch.io block reflects its role in the platform's distribution of the executable game, as well as the site where players must sign in to access and download the game.

**Figure 8**

*Sequence Diagram*

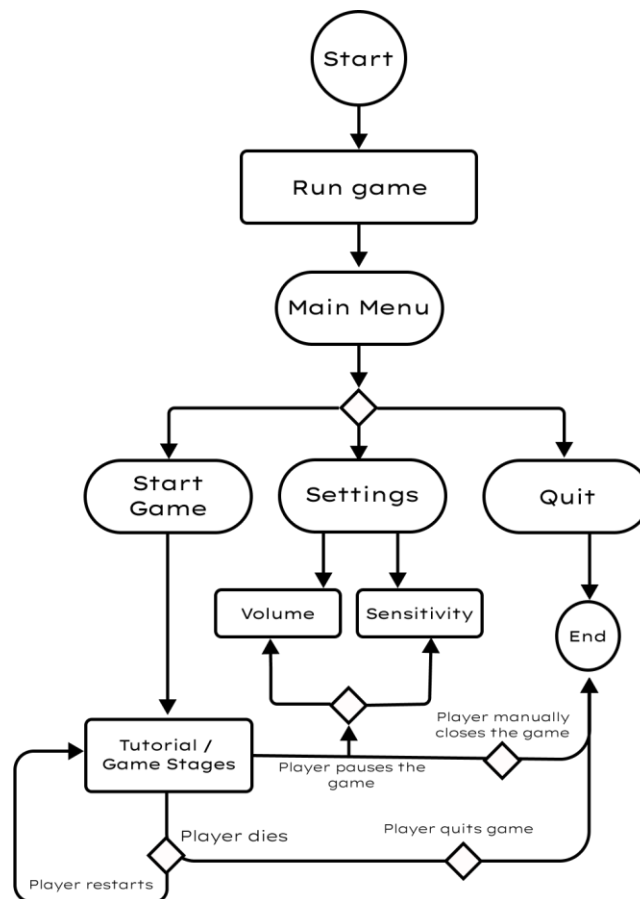


*Note.* Colored boxes represents the loop between the interaction of the enemy and main character.

The sequence diagram, as shown above, demonstrates how the player's movement and decision in the game will affect the enemy AI's response resulting to a combat interaction within the game. The colored boxes represent loops between the enemy-player interaction. Green represents the current stage, whether the player successfully finish a level or dies in the game; Violet represents the loop for the stage progress until the last level; and Blue represents the loop damage interaction of the player and enemy. The combat will end once the enemy deals damage and dies. The cycle will finish after every enemy has been defeated, at which point the player can go on to the following level until the last one.

**Figure 9**

*Activity Diagram*



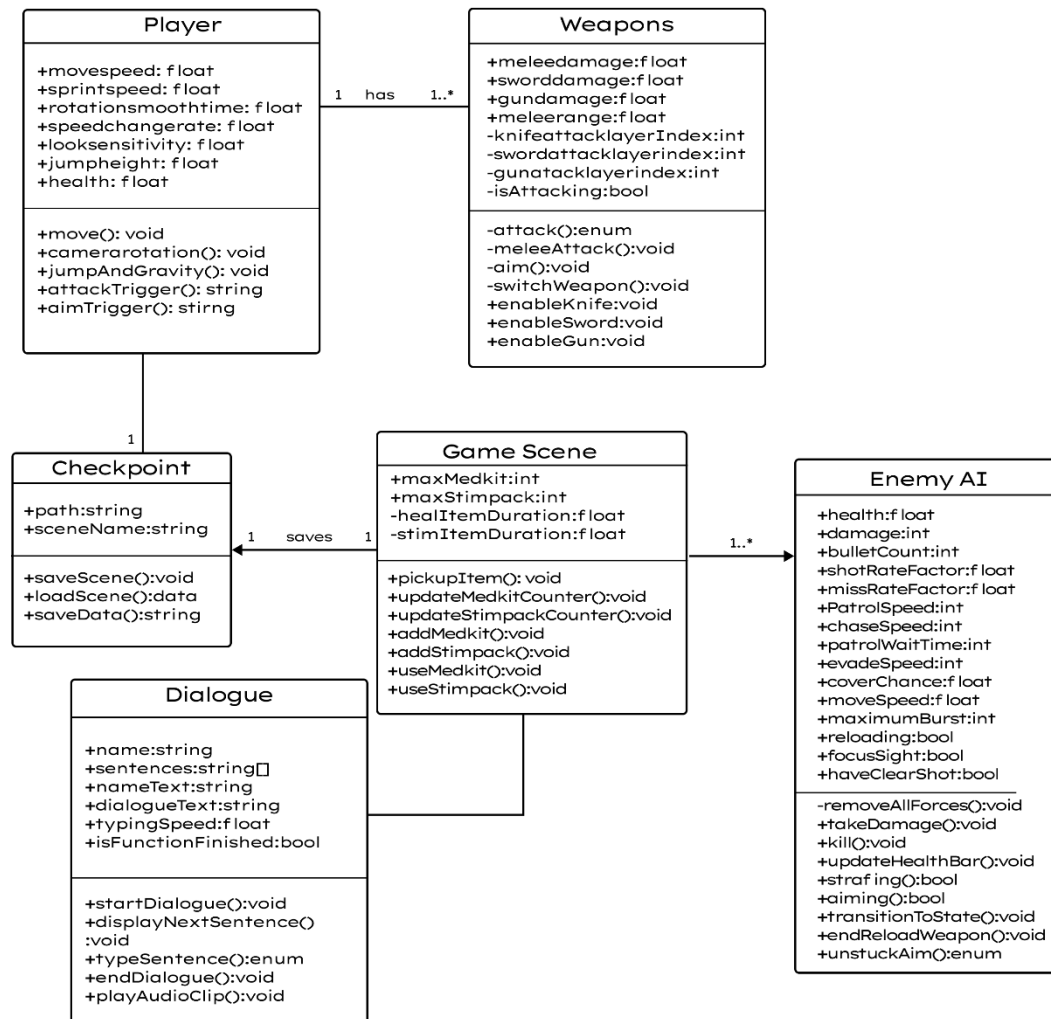
*Note.* Activity diagram of the game's basic functions

The activity diagram shown in the figure above shows the process of basic functionalities of the game application from the menus, to ending the game session.



**Figure 10**

*Class Diagram*





## Risk Assessment / Analysis

**Table 6**

*Risk Matrix in Bladescape's Game Development*

		impact		
		low	medium	high
likelihood	high	Toxic game addiction	compatibility conflict	unforeseen circumstances
	medium	Game piracy	bugs and errors	game plagiarism
	low	lack of fun factor to players	game accessibility	unexpected project quality issues

*Note.* The potential risks are categorized according to its likelihood and impact to the project.

The risk assessment matrix illustrated above is meant to help the project decrease the chance of probable risks to improve project performance. This matrix aims to provide a plan for mitigating risks that will impact the project. Risk is categorized by severity (red) and likelihood (green) using a color-coded approach. By using this matrix, it may lessen not just the severity of these risks for the project, but also their overall impact.





## **Design of Software Systems Products and/or Process**

### **Output and User-Interface Design**

The images below depict the game's user interface in all elements of interaction, from menus to gameplay. The set of figures are shown below from the preliminary template (prototype) of the game's UI to the actual UI in the published game.

#### **Figure 11**

*Main Menu (Initial Design)*



**Figure 11.1**

*Settings (Initial Design)*



**Figure 11.2**

*Paused Menu (Initial Design)*





**Figure 11.3**

*Settings - Paused State (Initial Design)*



**Figure 11.4**

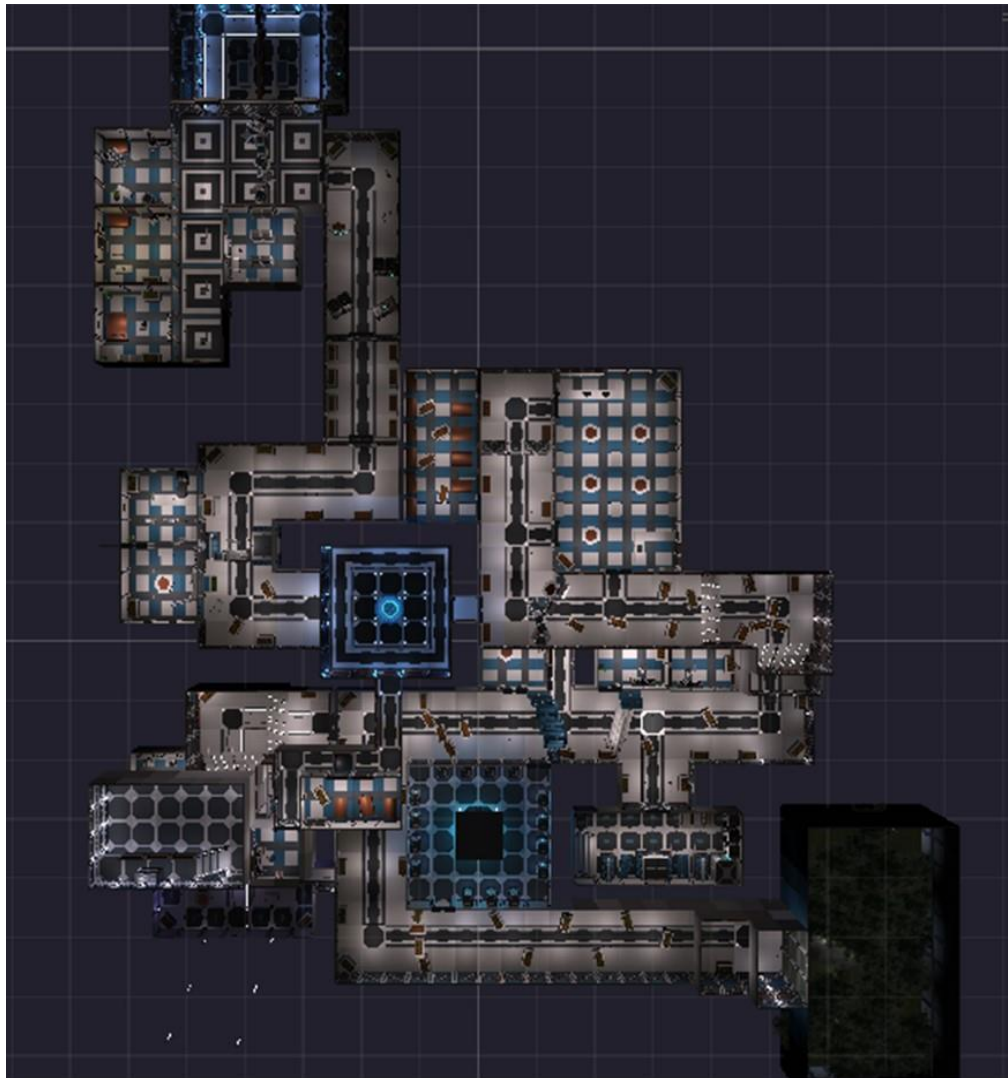
*In-game UI (Initial Design)*





**Figure 11.5**

*Bladescape Game Map (Top View)*



*Note.* The top view of the completed map of the game is divided into 4 different stages.



**Figure 12**

*Tutorial Map*



**Figure 13**

*Stage 1 Map*







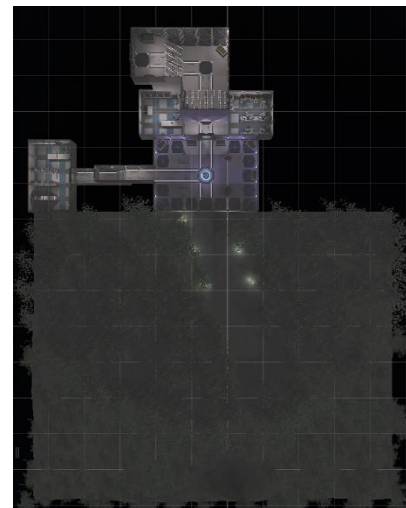
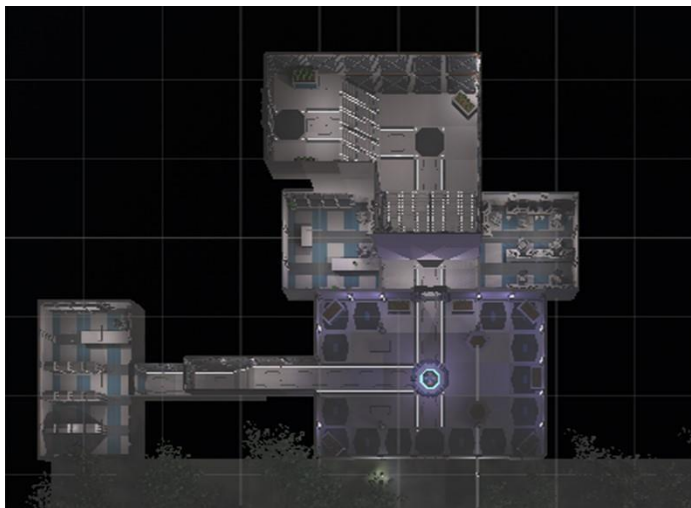
**Figure 13.1**

*Stage 2 Map Upper and Lower Level.*



**Figure 13.2**

*Stage 3 Map*





**Figure 14**

*Boss Fight Map*



**Figure 14.1**

*Main Menu (Actual Output)*





**Figure 14.2**

*Settings (Actual Output)*



**Figure 14.3**

*In-game UI (Actual Output)*

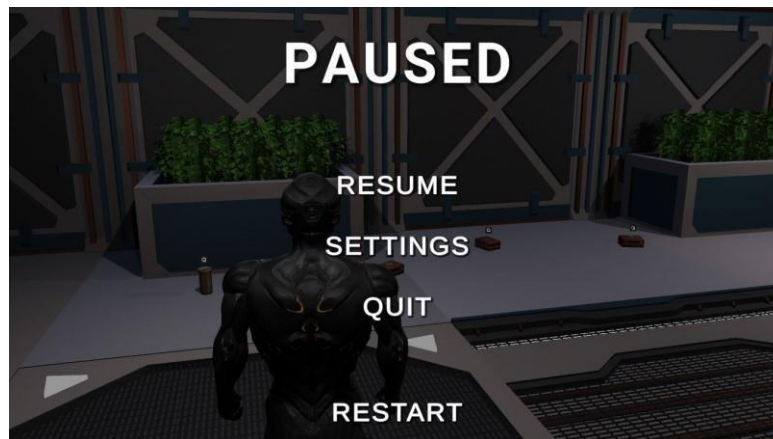






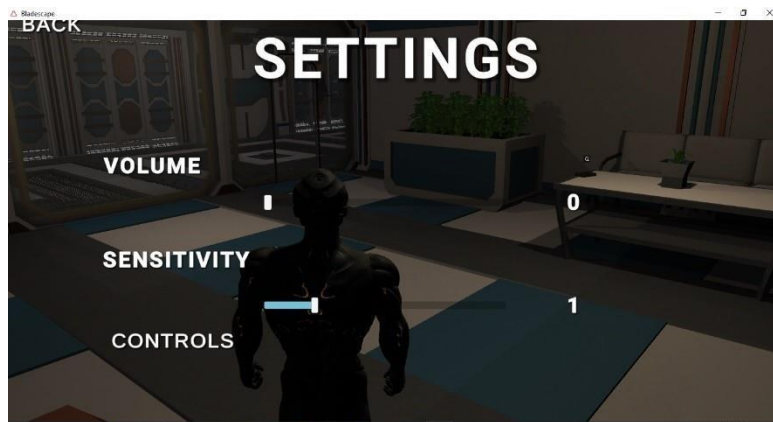
**Figure 14.4**

*Paused Menu (Actual Output)*



**Figure 14.5**

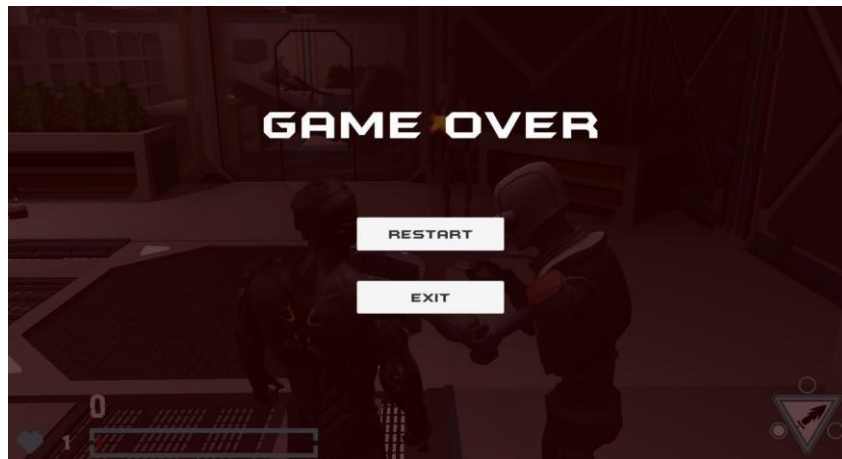
*Settings in Paused Menu (Actual Output)*





**Figure 14.6**

*In-game UI (Initial Design)*



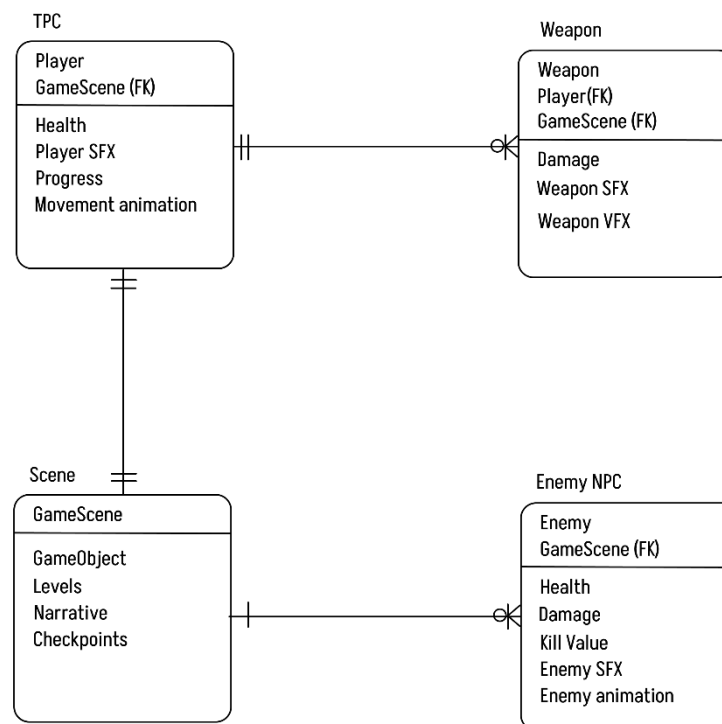


## Data Design

### Entity Relationship Diagram

**Figure 15**

*Entity Relationship Diagram*





## Data Dictionary

**Table 7**

### *Saved Progress Data*

Name	Data Type	Length	Scope	Purpose
maxhealth	const float	100f	global	variable for storing max health of player object
currenthealth	float	0f - 100f	global	variable for storing current active health of player object
medkitHealAmount	const int	35.0f	global	variable for storing healing applied to current player health
medkitHealDuration	const float	2.0f	global	variable for storing healing duration when player triggers heal function
medkitCount	int	0 to 4	global	variable for storing current number of healing items the player carries
stimpackCount	int	0 to 3	global	variable for storing current number of damage buff items the player carries
stimpackDuration	const float	10.0f	global	variable for storing duration of damage buff to player
stimpackDamageMultiplier	const float	1.1f	global	variable for storing a 10% damage buff to player once damage buff is active
defaultDamage	float	null	local	variable for storing a pre-buffed damage of player
animator	Animator	null	local	variable containing animator component of player
knife	GameObject	null	local	variable containing gameobject knife



sword	GameObject	null	local	variable containing gameobject sword
gun	GameObject	null	local	variable containing gameobject gun
knifeAttackLayerIndex	int	0 to 4	local	variable storing the animation index of Knife Attack Layer
swordAttackLayerIndex	int	1 to 4	local	variable storing the animation index of Sword Attack Layer
gunAttackLayerIndex	int	2 to 4	local	variable storing the animation index of Gun Attack Layer
attackTriggerName	string	0 to 50	local	variable storing the trigger parameter of melee attack animations
aimTriggerName	string	1 to 50	local	variable storing the trigger parameter of aim and shoot animations
knifeHitSound	AudioClip	null	local	container for audio clip component for knife attacks
swordHitSound	AudioClip	null	local	container for audio clip component for sword attacks
pistolShootSound	AudioClip	null	local	container for audio clip component for gun attacks
attackRadius	const float	1.0f	global	variable containing melee attack radius
meleeDamage	const float	10.0f	global	variable containing melee knife attack damage inflicted to enemies
swordDamage	const float	15.0f	global	variable containing melee sword attack damage inflicted to enemies
isAttacking	bool	0 to 1	local	boolean value to check if player is already attacking with equipped weapon



**Table 8**

*Third Person Controller Class Variables*

Name	Data Type	Length	Scope	Description
MoveSpeed	float	2	Global	move speed in character in m/s
SprintSpeed	float	5.335	global	sprint speed of the character in m/s
RotationSmoothTime	float	0.0 - 0.3	global	How fast the character turns to face movement direction
SpeedChangeRate	float	10	Global	acceleration and deceleration
FootstepAudioVolume	float	0.5	Global	sfx with world state interaction
LandingAudioClip	AudioClip	n/a	Global	sfx with world state interaction
FootstepAudioClips	AudioClip	n/a	Global	sfx with world state interaction
JumpHeight	float	1.2	Global	the height the player can jump
Gravity	float	15	Global	the character uses its own gravity value
JumpTimeout	float	0.5	Global	time required to pass before being able to jump again
FallTimeout	float	0.15	Global	time required to pass before entering the fall state
Grounded	boolean	true/false	Global	checks if character is on the ground
GroundedOffset	float	-0.14	Global	rough ground
GroundedRadius	float	0.28	Global	radius of the ground check
GroundLayers	LayerMask	n/a	Global	layers the character uses as ground



**Table 8.1**

*Third Person Controller Class Variables (Continuation)*

Name	Data Type	Length	Scope	Description
CinemachineCameraTarget	GameObject	n/a	Global	camera that follows the target for the Third POV
TopClamp	float	70	Global	maximum degrees of upward position of camera
BottomClamp	float	-30	Global	maximum degrees of downward position of camera
CameraAngleOverride	float	0	Global	fine tuning camera position when locked
LockCameraPosition	boolean	true/false	Global	for locking the camera position on all axis
_cinemachineTargetYaw	float	n/a	private	variable holder for yaw control of camera
_cinemachineTargetPitch	float	n/a	private	variable holder for pitch control of camera
_speed	float	n/a	private	variable holder for player speed
_animationBlend	float	n/a	private	variable holder for animation weight blending
_targetRotation	float	0	private	variable holder for target rotation
_rotationVelocity	float	n/a	private	variable holder for the player's rotation on y axis
_verticalVelocity	float	n/a	private	variable holder for the player's rotation on xz axis



**Table 8.2**

*Third Person Controller Class Variables (continuation)*

Name	Data Type	Length	Scope	Description
_terminalVelocity	float	53	private	maximum velocity of player on all axis
_jumpTimeoutDelta	float	n/a	private	checks if the player is in jump-state animation
_fallTimeoutDelta	float	n/a	private	checks if the player is in fall-state animation
_animIDSpeed	int	n/a	private	holder for animation weight in animation blending
_animIDGrounded	int	n/a	private	holder for animation weight in animation blending
_animIDJump	int	n/a	private	holder for animation weight in animation blending
_animIDFreeFall	int	n/a	private	holder for animation weight in animation blending
_animIDMotionSpeed	int	n/a	private	holder for animation weight in animation blending
_playerInput	PlayerInput	n/a	private	assigned for input direction
_animator	Animator	n/a	private	animator component for managing player animation state
_controller	CharacterController	n/a	private	managing 3D rigid body
_mainCamera	GameObject	n/a	private	holder for game scene main camera
_hasAnimator	boolean	true/false	private	checks if player model has animation states





**Table 9**

*Player Health Attributes*

Name	Data Type	Length	Scope	Description
maxHealth	int	0-100	global	variable that holds in-game value of player's max hp
currentHealth	int	0-100	global	variable that holds in-game value of player's current hp
healthBar	HealthBar	0-100	global	used to present player HP slider bar in 2D canvas
slider	Slider	n-a	global	to represent current hp in rectangle horizontal format
HPText	TextMeshProUGUI	0-100	global	used to represent hp in numerical format
gradient	Gradient	n-a	global	represents color change in hp value change
fill	Image	n-a	global	fill component of hp slider

**Table 10**

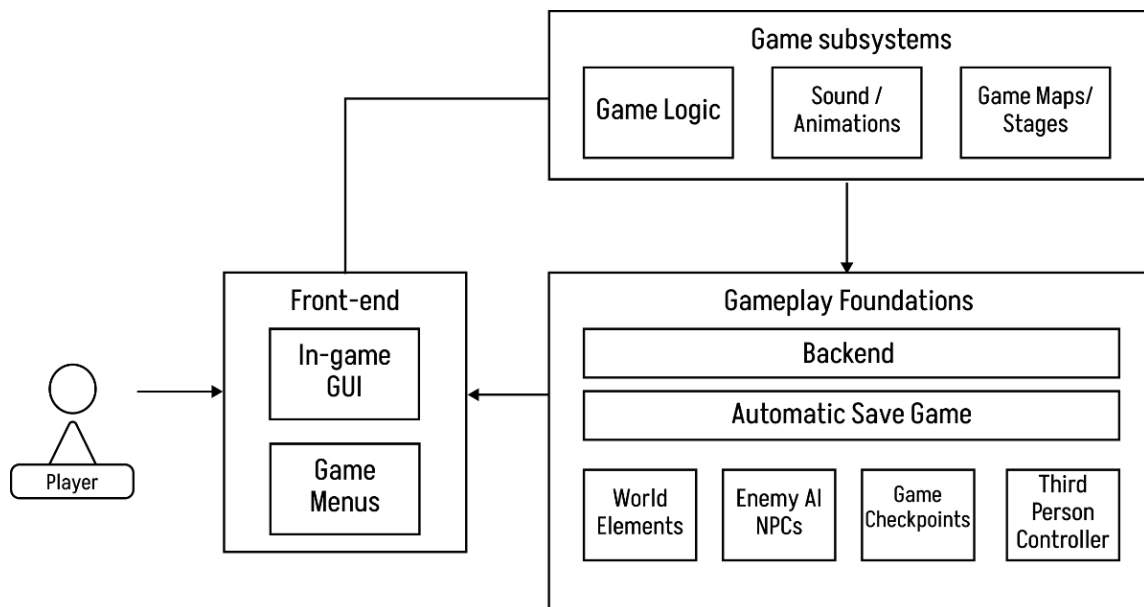
*Sound Variable Class*

Name	Data Type	Length	Scope	Description
name	string	100000	global	written string in editor to identify audio clip
clip	AudioClip	n-a	global	holder for imported audio file
volume	float	0-1	global	controller for in-game volume
pitch	float	0-1	global	controller for in-game volume
loop	boolean	true/false	global	variable intended to loop in-game audio (bgm, ambient, sfx)

## System Architecture

**Figure 16**

*Game Architecture of Bladescape*



The model shown above illustrates the three foundations that make up Bladescape varying from the front-end, foundations, the subsystems, and the progress saving system.

The front-end consists of the game's user interface starting from the menus and user interface.

The enemy NPC's behavior in the game is caused by the artificial intelligence in the game's subsystem. The game's levels are constructed from the game maps. The sound attributes are in charge of creating all of the gameplay's sound effects, including those created by player and opponent movement and animation, ambient sounds created by the environment, and background music. Lastly, the opponent and TPC combat scenarios are



created by the game's damage logic. The damage is calculated at this point to decide if the player has lost the game or will move on to the next one.

The game foundations are the essential elements of the game. Bladescape will not function correctly and will not satisfy the functional requirements without these components. The majority of its foundation is found on its backend. The automatic saving system is responsible for the progress saving functionality of Bladescape. By determining the most recent checkpoint reached by the player, the progress under that checkpoint is saved and will be synced online. The world elements consist of the game maps from the game subsystem block. The third person controller is the main character controlled by the player in the game. It is responsible for receiving inputs from the player using the keys to control its movement such as walking, running, and attacking.

### **Security**

As Bladescape is distributed through the platform that the proponents have selected, they guarantee that no harmful data will be collected from users who download the program from the internet. The variables and information explicitly collected from the application's progress values are the only things included in the saved progress, which does not include sensitive information about the users.

In addition, alterations of saved progress are not made possible due to translating the saved progress into hex values. This way, the users are prevented from cheating in the game.



## DEVELOPMENT AND TESTING

### Development

#### Software Specification

The software requirement specifications for the game application's development and implementation are defined out in this section.

**Table 11**

*Software Specification*

<b>Development</b>	<b>Game Engine</b>	2021.3.19f1 Unity Editor 2021
	<b>3D Editing Tools</b>	Unity 3D Editor / Blender latest version
	<b>Text Editing Tool</b>	Visual Studio Code 2022 Community Edition
	<b>Image Editing Tool</b>	Adobe Photoshop CS6 or higher
	<b>Video Editing Tool</b>	Adobe Premiere Pro
<b>Implementation</b>	<b>Operating System</b>	Microsoft Windows 10



## Hardware Specification

The hardware requirement specifications for the game application's development and implementation are defined out in this section.

### DEVELOPMENT

CPU:	6-Core Modern CPU
Storage:	500GB+ Solid State Drive
Memory:	32GB RAM
Video Memory:	at least 6GB
Operating System:	Windows 11 (64-bit)
Input/ Output devices:	Keyboard and mouse, headset
Network connectivity:	Wired Ethernet or Wi-Fi
Display:	at least 1920 x 1080p resolution

### USER REQUIREMENTS

CPU :	4-Core Modern CPU
Storage:	250GB+ Solid State Drive
Memory:	8GB RAM
Video Memory:	at least 2GB
Display:	at least 1080 x 720p resolution
Operating System:	Windows 10 (64-bit)
Network connectivity:	Wired Ethernet or Wi-Fi
Input/ Output devices:	Keyboard and mouse, headset



## **Program Specification**

Bladescape is an application that can be acquired from the itch.io, a marketplace for games created by independent developers. The game can also be acquired directly from the proponents. The game gives entertainment to its users by implementing AI NPCs within the game and a game story to drive the motivation of the player to finish the game.

## **Programming Environment**

### **Front End**

In creating the game application, the game assets, GUI, and menus are made using the Unity Editor and Visual Studio Code Community Edition. The researchers gathered most of the game assets from the Unity Asset Store and is modified using Blender and Unity Editor to make the assets cohesive to the game's concept and environment.

### **Back End**

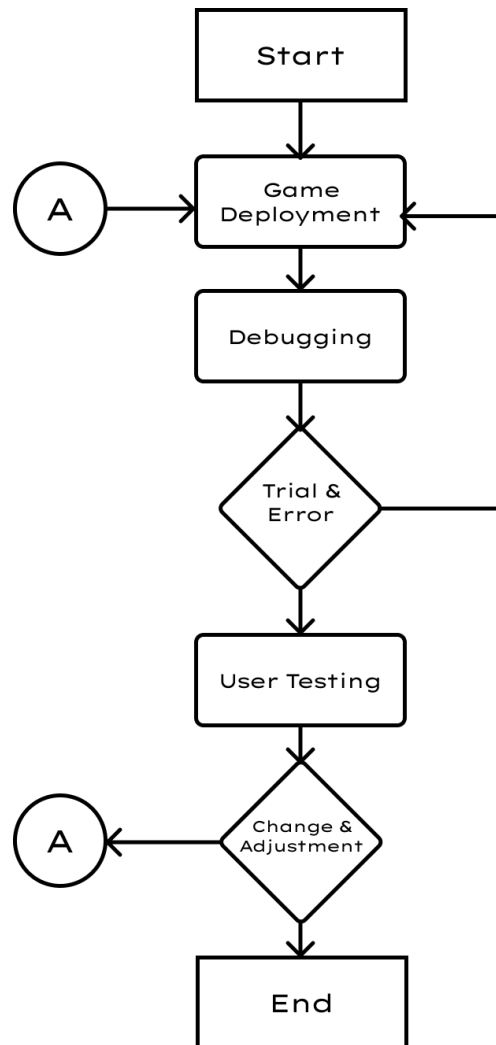
In the development of Bladescape, the proponents used C# as its backend's primary programming language. C# or pronounced as C-Sharp, is an object-oriented programming language that aims to combine the computing power of C++ with the programming simplicity of Visual Basic. C# is today's well-known backend language for games.



## Deployment Diagram

**Figure 17**

*Deployment Diagram*





## **Test Plan**

Bladescape: Single-player Action Game is a game application that will be used by the intended users. To guarantee that the application functions as planned, the proponents used a separate game scenes where they will test first a functionality before implementing it onto the actual output in the game application. This is to reduce bugs and errors when attempting to run the game.

The game scene for testing is named “Test Environment” and is used as a replica or a scratch to develop the game functionalities. If a certain function works on the test environment, the developers will then implement the working function to the final game scene where the final output of the game is executed.

In the testing plan, the proponents have decided to use unit testing and integration testing to check the application’s functional requirements.

## **Testing**

Testing is essential in this study as it helps to provide verification and validation on whether all of the functional requirements are met. This is done by generating test data and feeding it to the application to monitor the behavior. This helps to identify the vulnerabilities and limitations of the game application.





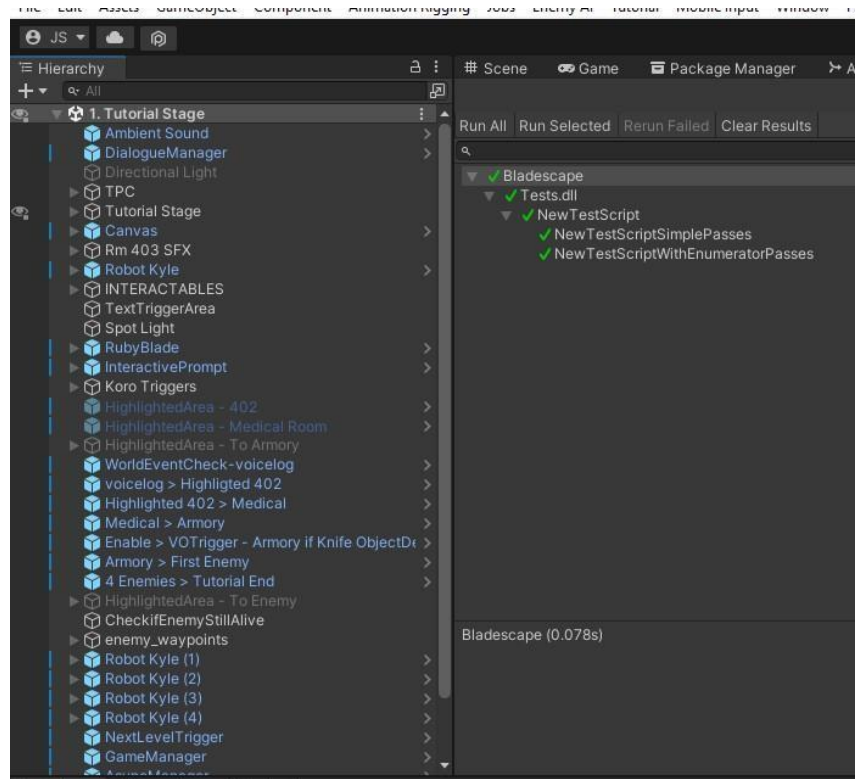
## Unit Testing

Unit testing is carried out by the proponents, particularly the developer, to ensure that the section of the application has accomplished the objectives of each function and to test whether they are operating as intended and producing the desired results.

In the unit testing process, the proponents used the Unity Test Framework, formerly Unity Test Runner, to test the codes both in edit and play mode. The testing process were divided into game scenes to test if the functions in each scenes are all working as intended.

**Figure 18**

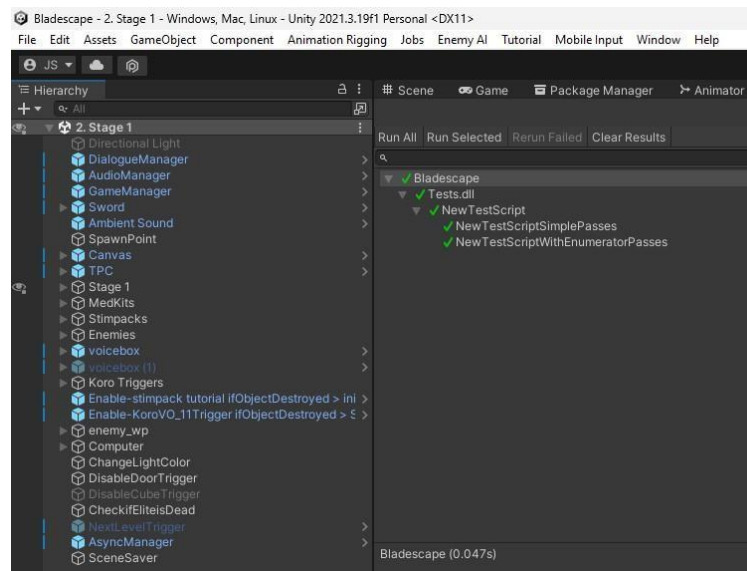
*Unit Testing Tutorial Stage*





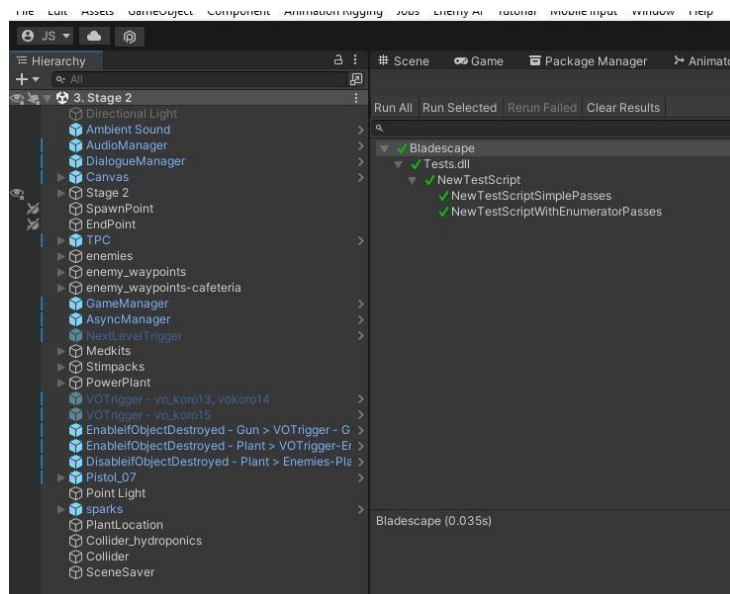
**Figure 18.1**

*Unit Testing Stage 1*



**Figure 18.2**

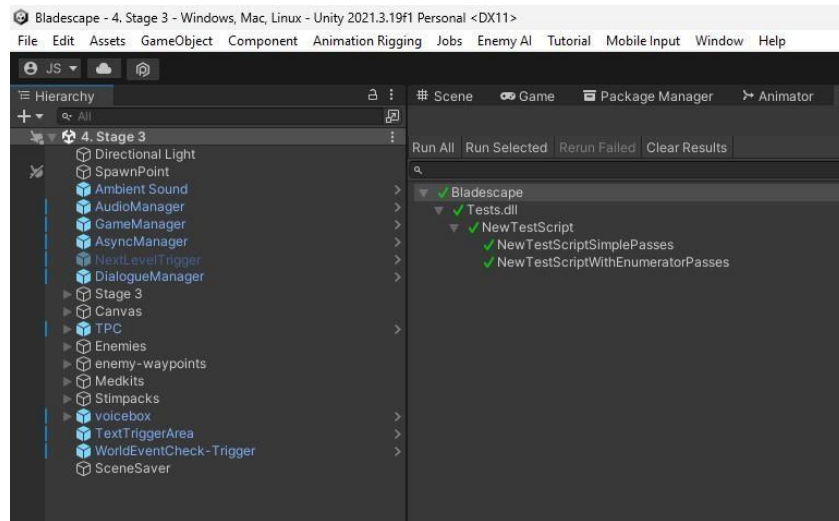
*Unit Testing Stage 2*





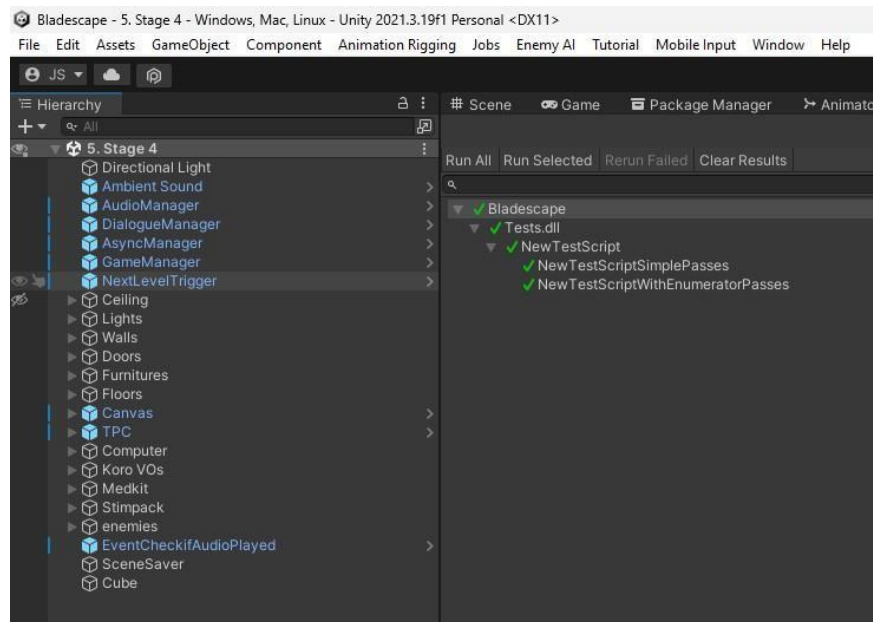
**Figure 18.3**

*Unit Testing Stage 3*



**Figure 18.4**

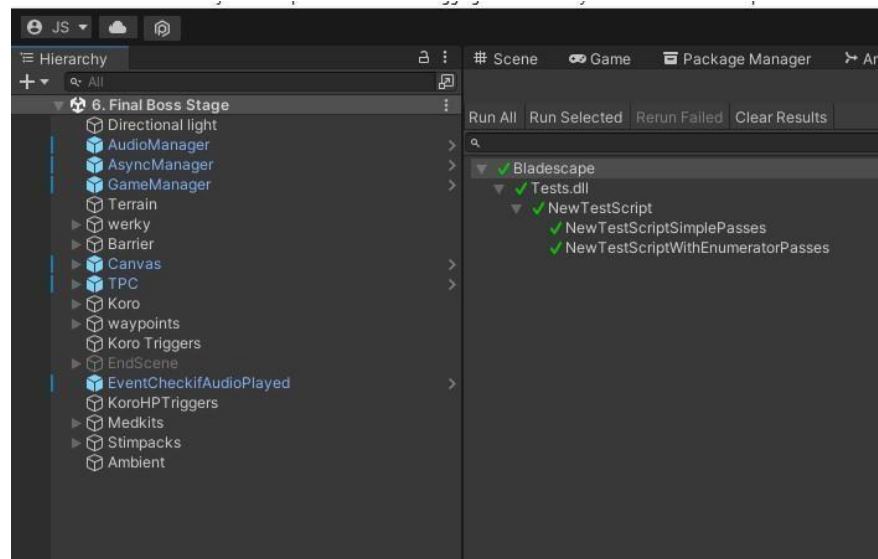
*Unit Testing Stage 4*





**Figure 18.5**

*Unit Testing Boss Stage*





## Integration Testing

Integration testing will be conducted after unit testing. Each function will be integrated to operate as a single unit during integration testing, producing the appropriate output as defined by each function's requirements.

In the process of integration testing, the proponents conducted a performance test using the MSI Afterburner to measure the frames rendered while running the game application.

## Performance Testing

### Figure 19

#### *Performance Testing Using MSI Afterburner*

```
02-05-2023, 01:04:16 Bladescape.exe benchmark completed, 17616 frames rendered in 304.172 s
Average framerate : 57.9 FPS
Minimum framerate : 0.1 FPS
Maximum framerate : 62.5 FPS
1% low framerate : 0.1 FPS
0.1% low framerate : 0.1 FPS
```

```
System Specs:
CPU: AMD Ryzen 5 5600x 6-core Zen 3|
GPU: Nvidia Geforce RTX 3060ti 8GB VRAM GDDR6
RAM: 32gb 3200mhz HyperX
OS: Windows 11 Build 22624
```

```
02-05-2023, 03:06:16 Bladescape.exe benchmark completed, 18125 frames rendered in 303.938 s
Average framerate : 59.6 FPS
Minimum framerate : 43.8 FPS
Maximum framerate : 62.0 FPS
1% low framerate : 12.2 FPS
0.1% low framerate : 7.5 FPS
```

```
System Specs:
CPU: AMD Ryzen 5 3600x 6-core
GPU: Nvidia Geforce GTX 1650ti 4GB VRAM GDDR6
RAM: 16gb 3200mhz Kingston Fury
OS: Windows 10
```



### **Alpha Testing**

Alpha testing is a kind of operational testing performed at the end of the proponents by a team of play testers or a potential user. Bladescape will go through alpha testing with its target users to ensure that it meets all of its functional and non-functional demands. Furthermore, to determine whether the suggested system would achieve the desired results.

### **Beta Testing**

Beta testing will come after alpha testing. This is done to determine the input from the recommended application. This technique will be carried out to evaluate the application's faults and defects, as well as if the system requires revision. Beta testing also allows you to see what users and developers are anticipating straight immediately.

The researchers will choose 20 participants for a closed beta test of the gaming application in this study. The proponents make use of the game's offline-based saving functionality while the game program is being prepared for release in the selected distribution platform.

Using the evaluation form and testing-related observations, the user experience during beta testing is analyzed. The results are described in the Chapter 5 of this study.



### **Description of the Prototype**

The goal of this capstone project is to create an entertaining single-player action game using AI-controlled non-playable characters. In Unity Editor, the basic prototype is made, and Figma is used for wireframing. However, in order to create a fully working application, resources must be gathered, much as the game's assets.

This led to the creation of static but realistic prototypes of a feature's design and intended use. The mockup acted as a prototype for this project. As a consequence, the prototype could be put to the test by the target market in real-world scenarios, teaching the developers things they hadn't previously considered. When developing anything, a prototype is quite important since it forces designers to think about fresh problems when converting a concept into something tangible.



### **Program Properties**

As the primary piece of software utilized by the development team, Unity Editor and 3D Unity Editor, and Visual Studio Code 2022 are included in the program's properties. Both the front-end and back-end properties are created using a computer with Microsoft Windows OS installed.

### **Function Properties**

A single-player action game with AI NPCs is what the game program is designed to provide. As a result, the majority of system requirements largely rely on application output. The output of images displayed on a user's screen, the actions of the AI NPCs, and the player's controller within the game all affect how the program functions.





## Implementation Plan

### Project Implementation Checklist

**Table 12**

#### *Implementation Plan*

No.	Tasks	Priority	Status
1	Testing of the game application	High	Completed
2	Fixing bugs	High	Completed
3	Publish game	Medium	Completed
4	Performing a survey	High	Completed
5	Maintain the game by improving through user's feedback	Low	In Progress

### Implementation Contingency

There may be performance lags and crashes in the game depending on the user's device specification and game's behavior. There are several methods to avoid such issues, but the proponents did their utmost to make Bladescape playable on entry-level devices by making it user-friendly, efficient, and compatible.

### Infrastructure / Deployment

Initially, Bladescape is deployed with its offline-based saving feature. While the game is ready for the users to play offline, the proponents published the game on itch.io. Bladescape must be deployed using the process provided by the chosen distribution platform. In the process, it is assumed that the developers have created a fully functional single-player game that is accessible for the users to download.



## Chapter V

### CONCLUSION AND RECOMMENDATIONS

This is the final chapter of the capstone project, and as such, it will include an overall evaluation as well as information on how the project might be improved in the future. Finally, depending on the project's findings, proposals for the project's direction will be considered.

#### Presentation, Analysis, and Interpretation of Data

##### a. Functionality

This characteristic shows the extent to which the game delivers functionalities that fulfill stated and implied demands when employed under defined situations.

**Table 13**

*WM and VI of the Functionality Evaluation*

Questions	4	3	2	1	Mean	WM	VI
Q1	12	8	0	0	3.6	3.6	Strongly Agree
Q2	13	7	0	0	3.65		
Q3	11	9	0	0	3.55		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The data from user experiences during game play are interpreted in the table above, focusing on the game's Functionality. With a weighted mean of 3.6 and a verbal interpretation of Strongly Agree based on the four-level Likert scale, the collection of questions focuses on the game's functionality.



### b. Reliability

This characteristic denotes how well a system performs defined functions under given conditions during a specific time period.

**Table 13.1**

*WM and VI of the Reliability Evaluation*

Questions	4	3	2	1	Mean	WM	VI
Q1	6	13	0	1	3.2	3.5	Strongly Agree
Q2	11	8	1	0	3.50		
Q3	16	4	0	0	3.8		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The facts from user experiences are interpreted in the table above focusing on the game's Dependability. The collection of inquiries focuses on the game's dependability and has a verbal interpretation of Strongly Agree based on a four-level Likert scale with a weighted mean of 3.5.



### c. Usability

This characteristic describes the degree to which a system may be utilized to achieve defined goals with effectiveness, efficiency, and satisfaction in a given context of usage.

**Table 13.2**

*WM and VI of the Usability Evaluation*

Questions	4	3	2	1	Mean	WM	VI
Q1	15	5	0	0	3.75	3.8	Strongly Agree
Q2	16	4	0	0	3.80		
Q3	15	5	0	0	3.75		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The aforementioned table illustrates how data from players' gameplay experiences were interpreted, focusing on the game's Usability. The collection of questions focuses on the usability of the game and has a verbal interpretation of Strongly Agree based on a four-level Likert scale with a weighted mean of 3.8.



#### d. Efficiency

This characteristic reflects the efficiency in relation to the quantity of resources under various circumstances.

**Table 13.3**

*WM and VI of the Efficiency Evaluation*

Questions	4	3	2	1	Mean	WM	VI
Q1	10	10	0	0	3.50	3.5	Strongly Agree
Q2	10	10	0	0	3.50		
Q3	12	8	0	0	3.60		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The data from user experiences during game play are interpreted in the table above, which highlights the Efficiency of the game. Based on a verbal interpretation of Strongly Agree using a four-level Likert scale, the collection of questions has a weighted mean of 3.5 and focuses on the game's efficiency.



### e. Portability

This characteristic reflects how well the game or program performs regardless of how the device is used.

**Table 13.4**

*WM and VI of the Portability Evaluation*

Questions	4	3	2	1	Mean	WM	VI
Q1	16	4	0	0	3.80	3.7	Strongly Agree
Q2	8	11	1	0	3.35		
Q3	16	4	0	0	3.80		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The above table displays an analysis of user experience data that focuses on the portability of the game. The collection of questions focuses on the portability of the game and has a verbal interpretation of Strongly Agree based on a four-level Likert scale, with a weighted mean of 3.7.



#### **f. Maintainability**

This characteristic demonstrates the degree of effectiveness and efficiency with which a system may be altered to enhance, rectify, or adapt to environmental changes and user needs.

**Table 13.5**

*WM and VI of the Maintainability Evaluation*

Questions	4	3	2	1	Mean	WM	VI
<b>Q1</b>	11	8	1	0	3.50	3.4	Strongly Agree
<b>Q2</b>	12	5	1	2	3.35		

*Note.* WM is Weighted Mean and VI is Verbal Interpretation

The table above shows the interpretation of data from users experience while playing the game showing focusing in the Maintainability of the game. The set of questions focuses on Maintainability of the game with a weighted mean of 3.4, having a verbal interpretation of Strongly Agree based on the four-level Likert scale.



## **Summary of Findings**

As a result of the respondents' evaluations of Bladescape: Single-Player Action Game, the researchers calculated a weighted mean of 3.6, which can be translated into Strongly Agree in terms of the quality standard of Functionality, Reliability, Usability, Efficiency, Portability, and Maintainability. The mean scores for each of these criteria were 3.6, 3.5, 3.8, 3.5, 3.7, and 3.4, respectively.

## **User Experience Results**

Gamers who own computers or laptops and IT professionals with experience reviewing games are the categories of users that took part in the survey evaluation. Out of 20 evaluators that completed the whole gameplay and survey, 85% were gamers and 15% were IT professionals. The gamers are categorized based on their skills in gaming (Junior 30%, Mid 40%, Advance 30%).

It is possible to draw the following conclusions from the observation made during the beta testing and the findings of the survey evaluation:

### **1. Functionality**

According to the evaluators' responses to the questions pertaining to the game's degree of functionality, the evaluators responded that the features, controls, and game mechanics performed as intended while being used.





## 2. Reliability

According to the evaluators' responses to the questions pertaining to the game's degree of reliability, the evaluators responded that the game is free of crashes and maintained a consistent level of performance throughout the gameplay. The saving function worked accurately and reliably.

## 3. Usability

According to the evaluators' responses to the questions pertaining to the game's degree of usability, the evaluators responded that the game is intuitive and easy to navigate in terms of the interface, and that the controls, tutorial, and game mechanics and narrative is easy to understand.

## 4. Efficiency

According to the evaluators' responses to the questions pertaining to the game's degree of efficiency, the evaluators responded that the game performed well on their device and did not experienced major performance issues. The game did make efficient use of their device's resources (e.g., RAM, GPU, battery).

## 5. Portability

According to the evaluators' responses to the questions pertaining to the game's degree of portability, the evaluators responded that the game was free of compatibility issues, hence, working well on their device.



## 6. Maintainability

According to the evaluators' responses to the questions pertaining to the game's degree of maintainability, the evaluators, particularly the gamers, responded that the game is easy to maintain and update. Some IT practitioners disagreed that the game is easy to maintain. Additionally, the majority of the respondents stated that they would want to see certain features or personalization choices added to the game. Some respondents said they had no features they would want to see added to the game.

## Conclusion

The proponents of Bladescape concluded that the game application satisfies the criteria for functionality, reliability, effectiveness, usability, portability, and maintainability based on the results and implementation of the application and testers' evaluation in the beta testing of Bladescape.

1. Given that the proponents used the software and hardware specifications specified, players of Bladescape played through the game without experiencing any major issues on their personal computers / laptops.
2. Before Bladescape is ready for beta testing, the release of the beta version of the game is greatly anticipated by the players, especially the evaluators, resulting to a fast data gathering and finding of respondents.



3. The game experience observed from the evaluators showed positive feedback on the game's overall quality in terms of its Functionality, Reliability, Usability, Efficiency, Portability, and Maintainability.
4. The method that the proponents used (creating separate game scene for the test environment and prototype) made the development progress slower but it reduced the bugs and errors encountered when running unit testing and integration testing.
5. Bladescape successfully incorporates artificial intelligence with the NPCs behavior resulting to positive feedback on gaming experience from the users. The AI NPCs also carried out fundamental tasks like patrolling, and attacking the places it is tasked with.
6. The game is successfully uploaded on itch.io, the proponent's chosen distribution platform. With this, Bladescape is made widely available to users after being published on the said website.



## **Recommendations**

Based on the findings and conclusion of the study, the following recommendations are encouraged:

1. The researchers recommend using Bladescape: Single-player Action Game to motivate future students of information technology at Rizal Technological University to build their own games by using this project as a reference or inspiration for the idea that student-made 3D games are feasible.
2. Bladescape used free assets that came with utilizing Unity Engine as its game engine, therefore the researchers advise using alternative tools to make their own assets and design their own character to match what is envisioned.
3. Given that Bladescape employs AI NPCs as opponents in the game, the researchers recommend the future researchers build a game that also incorporates AI NPCs as the players' ally or partner in the game.
4. The researchers recommend to the future researchers strive to create a multiplayer game in which two or more people can interact within the game while connected to the internet.
5. Bladescape is published and can be downloaded for free because it used free resources that limit its capability to be monetized. However, given enough time and money, this project has the potential to compete with other marketable games; consequently, the researchers recommend developing an ambitious, yet simple 3D



game based on this research to make a better version of a game developed by the researchers.

6. As Bladescape is only available for download on a chosen distribution platform, the researchers advise developers to explore and publish their games on other platforms so that they can reach a wider variety of audiences.



## BIBLIOGRAPHY

- Arjoranta, J. (2019, December). How to Define Games and Why We Need to. *The Computer Games Journal*, 8, 109-120.  
<https://doi.org/https://doi.org/10.1007/s40869-019-00080-6>
- Baker, T., Smith, L., & Anissa, N. (2019, February). *Educ-AI-tion: Exploring the future of artificial intelligence in schools and colleges*. Retrieved September 26, 2022, from Nesta.org:  
[https://media.nesta.org.uk/documents/Future\\_of\\_AI\\_and\\_education\\_v5\\_WEB.pdf](https://media.nesta.org.uk/documents/Future_of_AI_and_education_v5_WEB.pdf)
- Buttice, C. (2022, May 4). *Your Gaming For Over 40 Years*. Retrieved May 8, 2023, from Technopedia: <https://www.techopedia.com/finite-state-machine-how-it-has-affected-your-gaming-for-over-40-years/2/33996>
- LU, L. (2020, May 8). *China Patent No. CN111111202A*. Retrieved May 10, 2023, from <https://worldwide.espacenet.com/patent/search/family/070503653/publication/CN111111202A?q=pn%3DCN111111202A>
- Nanou, E. (2021, December 11). *6 Differences Between Action Games and Adventure Games*. Retrieved September 2022, from MUO (Make Use Of): <https://www.makeuseof.com/action-games-adventure-games-differences/>
- Podhradský, A. (2008). *Advanced Source Code Editor for the SIT Language*. Retrieved October 2, 2022, from Informační systém Masarykovy univerzity: <https://is.muni.cz/th/n5uhq/bc.pdf>
- Rapaport, W. J. (2018, May 25). What is a Computer? A Survey. *Minds and Machines*, 8, 385-426. <https://doi.org/DOI:10.1007/s11023-018-9465-6>
- Sinicki, A. (2021, March 20). *What is Unity? Everything you need to know*. Retrieved September 26, 2022, from Android Authority: <https://www.androidauthority.com/what-is-unity-1131558/>



Study.com. (2023, March 17). *Computer Game Definition, Overview & History*. Retrieved from Study.com: <https://study.com/academy/lesson/computer-game-definition-overview-history.html>

Suits, B. (2022, March 14). What is a Game? *Philosophy of Science*, 34(2), 148 - 156. <https://doi.org/https://doi.org/10.1086/288138>

Togelius, J. (2022, June 21). *Levelling-up NPCs with AI*. Retrieved September 26, 2022, from modl.ai: <https://modl.ai/levelling-up-npcs-with-ai/>

YANG, J. (2021, June 1). *China Patent No. CN112870722A*. Retrieved May 10, 2023, from <https://worldwide.espacenet.com/patent/search/family/076040980/publication/CN112870722A?q=pn%3DCN112870722A>