

YPC INTERNATIONAL COLLEGE



Programme: BSc. (Hons) Multimedia Computing

Final Year Project

6500YPROJB

Lecturer: Ms Rosmiza

749691 B150181



YPC INTERNATIONAL COLLEGE

YOU ARE NOT ALONE

SEAH KA WAI

**BSC.(HONS) MULTIMEDIA COMPUTING
2018**

You Are Not Alone

SEAH KA WAI

A project report submitted in partial fulfil of the requirements for the degree of
BSC. (HONS) MULTIMEDIA COMPUTING

YPC INTERNATIONAL COLLEGE

2018

DECLARATION

I at this moment declare that project work in this thesis is my own except for quotations and summaries which have been duly acknowledging.

18 April 2018

SEAH KA WAI

B150181/749691

ACKNOWLEDGEMENT

I would like to express my most profound appreciation to all those who provided me with the possibility to complete this project. The special gratitude I give to my final year project supervisor, Ms Rosmiza, whose contributing in giving suggestions and encouragement and also helped me to coordinate my project especially in writing this thesis. Thanks to GameDev.TV team, especially Lecturer Ben Tristem, and Lecturer Samuel Pattuzzi, to teaching how to use and implement the Unreal Engine. Moreover, Lecturer Ben Tristem and Lecturer Michael Bridges to teaching me How to use Blender to create 3D Object, and animation. Thanks to the community of gamedevTV, Udemy and Facebook Unreal and Blender group, to help me to solve the problem and error in the system.

Not to forget the tester from the friends such as Chin Cee Coy, Muhammad Sayid Zaini, and to the whole world who willing to download the Game via official website (SKW-Studio.com). Because of all interested people download and play the game, and return the feedback via email from the official website. The game keeps improve and change to user need.

Last but not least, thanks to Unreal which provides live training online on their website or youtube and a Youtuber name as Mathew Wadstein to explain what is the meaning and the use of each method.

ABSTRACT

This thesis will concern about the development of “You Are Not Alone”, a PC platform design for survival horror game. The user requires a machine as Laptop or Desktop which have an appropriate specification on the devices such RAM at least 6GB, Storage requires 5GB, Graphic Card require low minimum range, or else is difficult to get the sweet graphic, materials and texture in the game for playing and enjoy. CPU same as the Graphic Card requirement. Because the game match between the AAA game is rank on AA game. The video game is electronic game involves user interaction on optical device interacts to generated visual content. In 1973 and 1974, FPS (First Person Shooter) genre was created and the game called Maze War. Survival horror is a subgenre of video games inspired by horror fiction focuses on survive how the character will be alive with the dangerous situation. Currently, the PCs will have more advanced, the evolution of the game started from pixelated graphics to 4K resolution. The graphic of now day game is making more realistic into real life. User to using PCs of now day are keep increasing, for those who like to play high graphic of the game will prefer on PC platform. The prime objective of “You Are Not Alone” is to improve the user bravery level, add new hobbies to the user which can join with friend’s conversation and more calmly when facing the same situation. Apparently, this game may not be enough to bring the user steps out the fear from a horror movie, dark places. However, this is the first step for those users scare about the fear and can challenge or compete with. The user will able reduce themselves weakness to become strengths.

TABLE OF CONTENTS

| | |
|--|-----|
| DECLARATION | ii |
| ACKNOWLEDGEMENT | iii |
| ABSTRACT | iv |
| TABLE OF CONTENTS | v |
| LIST OF TABLES | ix |
| LIST OF FIGURES | x |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Introduction | 1 |
| 1.2 Project Background | 1 |
| 1.3 Problem Statement | 2 |
| 1.3.1 The decline in the level of terror (Timing Estimated) | 2 |
| 1.3.2 Inappropriate of Sound Effect..... | 2 |
| 1.3.3 Unlimited Sources Supply | 2 |
| 1.3.4 User Interest in Horror subject..... | 3 |
| 1.4 Objective | 3 |
| 1.4.1 To increase the scary level..... | 3 |
| 1.4.2 To provide Sound Effect will follow the specific location | 3 |
| 1.4.3 To limited Supply Sources..... | 3 |
| 1.4.4 To set the User Interest to Every Level | 4 |
| 1.5 Scopes..... | 4 |
| 1.5.1 Project Scope | 4 |
| 1.5.2 User Scope | 5 |
| 1.6 Project Benefits | 6 |
| 1.6.1 The Game with Movie Style | 6 |
| 1.6.2 Reduce Stress | 6 |
| 1.6.3 Can Manage User Critical Thinking..... | 6 |
| 1.6.4 Increase Senses Level | 6 |
| 1.6.5 Users are More Courageous..... | 6 |
| 1.7 System Specification..... | 7 |

| | | |
|--|---|-----------|
| 1.7.1 | Hardware..... | 7 |
| 1.7.2 | Software | 8 |
| 1.8 | Project Schedule / Time Management | 10 |
| 1.9 | Summary Questionnaire Result..... | 12 |
| 1.10 | Conclusion..... | 20 |
| CHAPTER 2 LITERATURE REVIEW | | 21 |
| 2.1 | Introduction | 21 |
| 2.1.1 | Improving cognitive and reading skills..... | 22 |
| 2.1.2 | Motivating logical thinking process..... | 23 |
| 2.1.3 | Strengthening observational skills. | 23 |
| 2.1.4 | Acquiring basic and factual knowledge. | 23 |
| 2.1.5 | Enhancing the abilities of problem-solving and decision-making..... | 23 |
| 2.1.6 | Developing strategic planning. | 23 |
| 2.1.7 | Supporting spatial awareness..... | 24 |
| 2.2 | Research On The Existing System/Web Domain (Or Previous System)..... | 24 |
| 2.2.1 | Goal..... | 24 |
| 2.2.2 | Rules | 24 |
| 2.2.3 | Feedback | 24 |
| 2.2.4 | Participation | 24 |
| 2.2.5 | Slender: The Arrival | 25 |
| 2.2.6 | Outlast..... | 27 |
| 2.2.7 | Emily Wants To Play | 29 |
| 2.3 | Findings On The Existing System/Web Domain (Or Previous System) | 31 |
| 2.4 | Development Methodology Background | 32 |
| 2.4.1 | Waterfall Model | 35 |
| 2.4.2 | Spiral Model..... | 37 |
| 2.4.3 | V-Model..... | 40 |
| 2.5 | Development Tools Background..... | 43 |
| 2.5.1 | Development..... | 43 |
| 2.5.2 | Designing | 46 |
| 2.5.3 | Version Control (Backup)..... | 51 |
| 2.6 | Conclusion..... | 53 |
| CHAPTER 3 DEVELOPMENT METHODOLOGY | | 54 |

| | | |
|--|--|-----------|
| 3.1 | Introduction | 54 |
| 3.2 | Methodology Chosen | 55 |
| 3.2.1 | Comparison Methodologies (Waterfall, Spiral, V-Mode) | 55 |
| 3.2.2 | Waterfall Model | 57 |
| 3.3 | Conclusion..... | 64 |
| CHAPTER 4 SYSTEM ANALYSIS | | 65 |
| 4.1 | Introduction | 65 |
| 4.2 | Fact-Finding Methods | 66 |
| 4.2.1 | Observation | 66 |
| 4.2.2 | Questionnaires..... | 66 |
| 4.2.3 | Interviews..... | 67 |
| 4.2.4 | The method of Fact-finding used in the Project (Questionnaire) | 68 |
| 4.3 | System Requirement | 69 |
| 4.3.1 | Functional Requirement..... | 70 |
| 4.3.2 | Nonfunctional Requirements | 71 |
| 4.4 | Conclusion..... | 71 |
| CHAPTER 5 SYSTEM DESIGN | | 73 |
| 5.1 | Introduction | 73 |
| 5.2 | Modelling | 73 |
| 5.2.1 | Data Flow Diagram..... | 74 |
| 5.2.2 | Flowchart | 75 |
| 5.2.3 | Use Case Diagram..... | 77 |
| 5.2.4 | Activity Diagram | 79 |
| 5.3 | User Interface Design..... | 82 |
| 5.3.1 | Main Menu User Interface | 82 |
| 5.3.2 | Pause Menu User Interface | 83 |
| 5.3.3 | Options Menu User Interface..... | 84 |
| 5.3.4 | Buildings | 85 |
| 5.4 | System Structure | 87 |
| 5.5 | Conclusion..... | 88 |
| CHAPTER 6 IMPLEMENTATION AND PROGRAMMING SYSTEM | | 89 |

| | | |
|---|--|------------|
| 6.1 | Introduction | 89 |
| 6.2 | Module Implementation | 89 |
| 6.2.1 | System Screenshot (Code) | 90 |
| 6.2.2 | Game System Screenshot..... | 105 |
| 6.3 | Conclusion..... | 116 |
| CHAPTER 7 TESTING AND IMPLEMENTATION | | 117 |
| 7.1 | Introduction | 117 |
| 7.2 | Testing Level..... | 117 |
| 7.2.1 | Unit/Component Testing..... | 117 |
| 7.2.2 | Integration Testing | 122 |
| 7.2.3 | System Testing..... | 123 |
| 7.2.4 | User Acceptance Testing | 123 |
| 7.3 | Conclusion..... | 129 |
| CHAPTER 8 CONCLUSION | | 130 |
| 8.1 | Introduction | 130 |
| 8.2 | Outcome Summarization..... | 130 |
| 8.3 | Strength And Weaknesses Of Project | 133 |
| 8.3.1 | Strengths | 133 |
| 8.3.2 | Weaknesses | 134 |
| 8.4 | Suggestion For Enhancement/Future Planning | 135 |
| 8.4.1 | More Texture and Material for Low-end PC support | 135 |
| 8.4.2 | VR (Virtual Reality) experience | 135 |
| 8.4.3 | DLC (Downloadable Content)..... | 135 |
| 8.5 | Conclusion..... | 135 |
| REFERENCES..... | | 137 |
| APPENDICES | | 142 |
| LOGBOOK..... | | 163 |

LIST OF TABLES

| | |
|---|-----|
| Table 1. 1: Blender Requirement | 7 |
| Table 1. 2: Unreal Engine Requirement | 8 |
| Table 2. 1: Comparison of Existing Game..... | 30 |
| Table 2. 2: Major advantages and disadvantages of the waterfall model | 37 |
| Table 2. 3: Major advantages and disadvantages of Spiral model..... | 40 |
| Table 2. 4: Major advantages and disadvantages of V-Model..... | 43 |
| Table 3. 1: Comparison of Waterfall, Spiral, V-Model | 55 |
| Table 4. 1: Advantages and Disadvantages of Observation | 66 |
| Table 4. 2: Advantages and Disadvantages of Questionnaires. | 67 |
| Table 4. 3: Advantages and Disadvantages of Interviews | 68 |
| Table 7. 1: Unit Testing (Main Menu)..... | 119 |
| Table 7. 2: Unit Testing (Player) | 120 |
| Table 7. 3: Unit Testing (Enemy) | 121 |
| Table 7. 4: Unit Testing (Checkpoint) | 121 |
| Table 7. 5: Unit Testing (Game Scene) | 122 |
| Table 7. 6: Integration Testing | 123 |
| Table 7. 7: System Testing..... | 123 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1. 1: Time Management and Gantt Chart (part 1) | 10 |
| Figure 1. 2: Time Management and Gantt Chart (part 2) | 11 |
| Figure 1. 3: Time Management and Gantt Chart (part 3) | 11 |
| Figure 1. 4: 3D Game Development Questionnaire Result (part 1) | 12 |
| Figure 1. 5: 3D Game Development Questionnaire Result (part 2) | 13 |
| Figure 1. 6: 3D Game Development Questionnaire Result (part 3) | 13 |
| Figure 1. 7: 3D Game Development Questionnaire Result (part 4) | 15 |
| Figure 1. 8: 3D Game Development Questionnaire Result (part 5) | 16 |
| Figure 1. 9: 3D Game Development Questionnaire Result (part 6) | 17 |
| Figure 1. 10: 3D Game Development Questionnaire Result (part 7) | 16 |
| Figure 1. 11: 3D Game Development Questionnaire Result (part 8) | 19 |
| | |
| Figure 2. 1: Slender: The Arrival of Steam (Midnight City, Majesco Entertainment, 2013) | 25 |
| Figure 2. 2: Comment for Slender: The Arrival of Steam (Comment, 2013)..... | 26 |
| Figure 2. 3: Outlast from Steam (Red Barrels, 2013)..... | 27 |
| Figure 2. 4: Comment for Outlast from Steam (Comment, 2013)..... | 28 |
| Figure 2. 5: Emily Wants To Play from Steam (SKH Apps, 2015)..... | 29 |
| Figure 2. 6: Comment for Emily Wants To Play from Steam (Comment, 2015)..... | 30 |
| Figure 2. 7: Typical Software Development Life Cycle (SDLC, n.d.)..... | 33 |
| Figure 2. 8: Waterfall Model (Waterfall, n.d.)..... | 35 |
| Figure 2. 9: Spiral Model (Spiral, n.d.)..... | 38 |
| Figure 2. 10: V-Model (VModel, n.d.) | 41 |
| | |
| Figure 3. 1: Waterfall Model (Waterfall, n.d.)..... | 57 |
| Figure 3. 2: Best-Selling Video Game Genres in 2014 (Best Selling Video Game, 2014) | 58 |
| Figure 3. 3: Most Popular 3D Modeling Software 2017 (Most Popular 3D Modeling & Design Software for 3D Printing, 2017) | 60 |
| | |
| Figure 5. 1: Data Flow Diagram | 74 |

| | |
|--|-----|
| Figure 5. 2: Flowchart (page 1)..... | 76 |
| Figure 5. 3: Flowchart (page 2)..... | 77 |
| Figure 5. 4: Use Case Diagram..... | 79 |
| Figure 5. 5: Activity Diagram..... | 81 |
| Figure 5. 6: Main Menu UI..... | 82 |
| Figure 5. 7: Pause Menu UI..... | 83 |
| Figure 5. 8: Options Menu UI..... | 84 |
| Figure 5. 9: First Building (Abandoned Motel, first floor)..... | 85 |
| Figure 5. 10: First Building (Abandoned Motel, second floor) | 86 |
| Figure 5. 11: System Structure..... | 87 |
| | |
| Figure 6. 1: YouAreNotAloneCharacter Class | 90 |
| Figure 6. 2: YouAreNotAloneCharacter Class .cpp (Part 1) | 91 |
| Figure 6. 3: YouAreNotAloneCharacter Class .cpp (Part 2) | 92 |
| Figure 6. 4: BP Player (Running Event) | 93 |
| Figure 6. 5: BP Player (Crouch Event) | 94 |
| Figure 6. 6: BP Player (On/Off Light Event)..... | 94 |
| Figure 6. 7: BP Player (Switch Event)..... | 95 |
| Figure 6. 8: BP Player (Open Pocket Inventory Event)..... | 95 |
| Figure 6. 9: BP Player (Quick Check Event)..... | 95 |
| Figure 6. 10: BP Player (Damaged Trigger)..... | 96 |
| Figure 6. 11: BP Player (Healing Event) | 96 |
| Figure 6. 12: BP Player (Reload Event Part 1) | 97 |
| Figure 6. 13: BP Player (Reload Event Part 2) | 97 |
| Figure 6. 14: BP Checkpoint (Save Game Function) | 98 |
| Figure 6. 15: BP Player (Load Game Function) | 98 |
| Figure 6. 16: ABP Player with Additive (Anim Graph) | 99 |
| Figure 6. 17: ABP Player with Additive (Locomotion)..... | 99 |
| Figure 6. 18: ABP Player with Additive (Animation Event)..... | 100 |
| Figure 6. 19: ABP Player with Additive (Macro Footstep Song)..... | 100 |
| Figure 6. 20: BP Locker (Hide Event) | 100 |
| Figure 6. 21: BP AI Controller School Girl (Perception Event)..... | 101 |
| Figure 6. 22: BP AI Controller School Girl (Invisible Event)..... | 101 |

| | |
|--|-----|
| Figure 6. 23: BTTask AttackPlayer | 101 |
| Figure 6. 24: BT School Girl | 102 |
| Figure 6. 25: EQS School Girl..... | 103 |
| Figure 6. 26: ABP School Girl (Animation Graph)..... | 103 |
| Figure 6. 27: ABP School Girl (Locomotion) | 104 |
| Figure 6. 28: ABP School Girl (Animation Event)..... | 104 |
| Figure 6. 29: Level Blueprint – Night (Random Jump scare)..... | 104 |
| Figure 6. 30: Main Menu Screen | 105 |
| Figure 6. 31: Menu Option Screen..... | 106 |
| Figure 6. 32: Setting Option Screen (Display)..... | 106 |
| Figure 6. 33: Setting Option Screen (Graphic) | 107 |
| Figure 6. 34: Setting Option Screen (How To Play)..... | 108 |
| Figure 6. 35: Loading Screen..... | 108 |
| Figure 6. 36: Pause Menu | 109 |
| Figure 6. 37: Key and Note..... | 109 |
| Figure 6. 38: Pocket Key Inventory | 110 |
| Figure 6. 39: Phone flashlight and Hints | 110 |
| Figure 6. 40: Flash hlight and Blood effect | 111 |
| Figure 6. 41: Hide | 111 |
| Figure 6. 42: Game Over Screen..... | 112 |
| Figure 6. 43: Credit Scene | 112 |
| Figure 6. 44: Motel in Daytime..... | 113 |
| Figure 6. 45: Slum in Daytime..... | 113 |
| Figure 6. 46: Police Station in Daytime..... | 114 |
| Figure 6. 47: School in Daytime | 114 |
| Figure 6. 48: Hospital in Daytime..... | 115 |
| Figure 6. 49: Church in Daytime | 115 |
| | |
| Figure 7. 1: Questionnaire Feedback (part 1) | 125 |
| Figure 7. 2: Questionnaire Feedback (part 2) | 126 |
| Figure 7. 3: Questionnaire Feedback (part 3) | 127 |
| Figure 7. 4: Questionnaire Feedback (part 4) | 125 |

CHAPTER 1

INTRODUCTION

1.1 Introduction

A video game is an electronic game involves user interaction on the visual device to interact the generated visual content such as computer monitor or television. Any display devices that can produce two- or three-dimensional images have been indicated since the 2000s. In 1973 and 1974, the video game genre first-person shooter (FPS) was created and the game called Maze War. FPS is a video game genre centred on the gun or other weapon-based combat in a first-person perspective. Survival horror is a subgenre of video games that inspired by horror fiction that focuses on survival of the character as the game tries to frighten players with either horror graphics or scary ambience. Although combat can be part of the gameplay, the player is made to feel less in control than in typical action games through limited ammunition, health, speed and vision, or through various obstructions of the player's interaction with the game mechanics. The player is also challenged to find items that unlock the path to new areas and solve puzzles to proceed in the game. Those games make use of excellent horror themes, like dark maze-like environments and unexpected attacks from enemies.

1.2 Project background

The background of this project is a single player video game. The title of the game name is "You Are Not Alone". The game is horror type with involving a small part educational game criteria. The story of this video game is a brave group child in an old village; they found out a mystery place which stores an old toy board. When the game had been playing, the consequences could be disastrous. The main story of the game is the main character has grown up and decided to go back to the village because she had

received an old friend (the brave child) invitation. However, what the main character thought is different than what she faces. The game is moving toward a horror scene and based on our real life. To implement the majority of horror movie scary scene, sound, action and movement. It does not allow the player to attack or fight; it is more to survival. The main point of this game is about hiding and find out what had happened before and the solution to solve the problem. To make the gamer more in-depth into the game, it implements FPS (First Person Shooter) camera view, in the dark place with limitation lighting supply. The game will have multiple story end scenes; it depends on the player decision and strategy skill level. The game includes several educational parts after the player has played the game and will notice the aftermath that we are cannot underestimate.

1.3 Problem statement

In the project, we have seen two type of category about the problem statement. One is relating to the system such as movie or game, and another one is more to user problem.

1.3.1 The decline in the level of terror (Timing Estimated)

The first problem is the relating type of our project. People are preferred to watch and play horror type of game because of the jump scare moment. However, the timing of the jump scare scene in today's movies and games can be estimated by audiences. The jump scares just slide of images, and the scary level has dropped so much.

1.3.2 Inappropriate of Sound Effect

The second problem is about the sound effect. The horror game is incomplete without specific sound effects. However, some of the horror games are not good in the sound effect such as playing the footstep sound in wrong moments. The game will become unattractive and reduced the interest of the player. It will stop the user to keep playing the game until the end.

1.3.3 Unlimited Sources Supply

The final problem is the system category. Currently, there are much of the horror game will have unlimited or huge amount supply source to support the user easily completed the game. For example, a game of a light source (Flash light) have no battery drain problem. The game will become much easier for user finished the game however user is not much involved in any fear or funk.

1.3.4 User Interest in Horror subject

Next is the problem about specific users. In the present society, some of the audiences are not brave enough to watch any horror, scary or even bloody scene movie or film. There are also a group of people who does not dare to play any supernatural genre game as well. This may be the problem that causes them decrease the perception of the dangerous and risking level.

1.4 Objective

For the objective of the project, it will be solving out all the problem had been mention from 1.3.

1.4.1 To increase the scary level

The first problem is the jump scare. Nowadays, the movie, film and game mainly depend on jump scare scene. The jump scare moment plays an essential part in the horror genre. Because of the jump scare scene failed, it may affect the products population significantly. The main problem of the failed jump scare is caused by wrong timing or just by showing a single image. Our project will be implementing the jump scare appearing in randomise timing with a different type of movement, motion or animation and effect. Randomize timing with a different type of action will bring up the jump scare level to another scary level as well. To make the jump scare not just showing for a short moment, it will bring mixed feelings to a player such as tension, scary, nervous.

1.4.2 To provide Sound Effect will follow the specific location

The second problem is not allowing missing out the sound effect. A horror game without sound effect is just nothing. To ensure the game are deeply scary, our game will be assuring the sound effect are needed to set accordingly in specified locations. For example, the footstep sound is only allowed send to the audience left ear. No other sound effect will be overlapping the game scene; it will purely become background sound.

1.4.3 To limited Supply Sources

The third problem is the game have much supply for user. A horror game without limited supply sources make the game less of fun, and will reduce the strength of fear such as jumps care. To ensure the game are scarier and toward in real life, limited supply

sources and random places in these supplied will increase the difficulty level and realistic level.

1.4.4 To set the User Interest to Every Level

Next problem will be the minority audiences are not brave enough to watch or play any horror game, and it leads possible to decrease the perception of danger. Our game will be the best solution for the group of minority people step out their first step. Our game may possibly encourage the audience to deal with the horror movie in the future.

1.5 Scopes

1.5.1 Project Scope

The scope of our game and the primary purpose is providing a set of the story plot. The audience will understand and learn something from the game. Structured and unstructured information is provided. Our project “You are not Alone” is a single player horror game, it also involves educational criteria such as user decision will change the end of the story and the moral they are going to learn from the game. It is more towards on survival type horror game with limitation supplies. The game is available on the PC (Personal Computer) platform. The player will progress through each obstructed level which requires precise manipulation of the environment, understand the situation. Through the game encourage the audience to stand out and facing what they scare and make the horror game to the next trend. The game will need the user to collect each missing item to read and figure out the way to complete the game with either good ending or the secret ending scene. Each item such as note will be helpful in the game. The game is more of a survival type horror game; the player is only allowed to run and hide in specific hiding spot. The player is not able to attack. The player must also plan to use any of the resources in the game, such as a battery. Because without light sources, the higher the chance will get the jump scares. The high-level skill of development is required to create the game. Game story, level, object, animation, graphic, script, game engine facilities are what I am mainly focusing on. Before creating the game, the storyline is the central part of the whole game; Level makes the game to have several difficulties; Object is 3d Mesh object with animation. The graphic can be a user interface or the texture map such as diffuse, high, bump map and UV map to apply in the 3d mesh. The script can be scripted as gaming text or the coding script as well. Moreover, to be able implemented with the Game engine. The game will be the best indie horror game ever, and it will include VR (Virtual Reality) technology support in

future once third-party company fund in this game and has a possibility for the DLC (Extension pack) to release the game series. For the risk, it will be the time we faced to complete the project because every second can alter the whole project. This might also be one of the constraints as well. The project will be facing another constraint due to limited budget support. For the marketing site, it may be possible to deliver the game via the Official Website or Game Platform (Steam).

1.5.2 User Scope

In this project, the user scope has only one position. It is controlling the main character throughout the game. The player is allowed to control the character either standing as idle, walking, running or jogging. Each of this will have different properties; the properties are connecting and directing the message to the AI character. Those AI characters will have own perception to identify and to roam to check the suspended area. All of the AI characters' perception will be similar to a real-life human. Such as eye perception and hearing perception. The user can also control the character to crouch movement; it will lower the chance to let the AI character to notice it. In the game, the supplier is a limitation; the user can control the character and collect each supplier such as a battery to find a way and survive. To survive in this game, hiding is a right way to avoid the jump scare, and it will be completed quickly. The user can also hide the character in some right hiding spots similar to the real world as well, such as a wardrobe or under the bed. The game is only specified to allow user control the character to jump or vault, it is to avoid disruption of the game life cycle. Towards the end of the game, the user can decide in the last chapter of the game; the decision making will make the story have a different ending. It will categorise into three different endings, sad ending, good ending and secret ending. Except for the controlling character, the user is allowed to change the setting of the game such as enable subtitle or allowing display subtitle with specify language, volume setting, graphics and screen resolution through the User Interface menu.

1.5.2.1 Target Audience

The game does have an age restriction and the game rating as PG13. It can be played by any person that is up to 13 years old or with parental guidance. The reason why the game has to set the age restriction it because the game is containing some strong language, violence, suggestive themes, blood and adult situations is not suitable for younger children.

Teenagers and adult are targeted for this project. It may also attract the horror genre lover and the group of people that new to the horror genre.

1.6 Project benefits

From this game, how can it be attracted to users? Moreover, what are the benefits of playing the game? There are plenty of the reason and benefits that user will achieve by playing the game and learn to apply in real life.

1.6.1 The Game with Movie Style

How to attract the user and their interest? This game is more of a movie type of game. The storyline is also based on the real life that possibly happened before. The user will get more interest to play and try it out.

1.6.2 Reduce Stress

For this project, the primary target user is the minor group of persons are fear to step out the horror genre. However, another part of the target user is the person who's like to challenge and will like to play the horror game as they live. When the user is playing any game, want to reduce the stresses. Same as survival horror game, because the type of game will be much more challenging and need user spend much more time to figure out how to find a way to exit or understand the whole story. User much more focus and enjoying in the gameplay will have the best result to decrease the stresses.

1.6.3 Can Manage User Critical Thinking

The user will be able to manage their critical thinking skill which people are usually lacked in daily life. Because our game is horror type, it can train the user response quickly when they are in dangerous places.

1.6.4 Increase Senses Level

Nowadays, most of the people are unawareness whenever they in dangerous places. The user will gain the skills after playing the game; the user is not just focused on technology devices in public places. Every time they will be more aware of surrounding and able to reduce the risk of dangers.

1.6.5 Users are More Courageous

A minority of the people cannot accept the facts about supernatural or any relationship to the horror genre. Due to the reason, they scare walking alone in the dark, or seeing

blood may cause them fainted. To help the minority people in the society, our game will boost their level of courageous and able to help them face reality.

1.7 System specification

1.7.1 Hardware

| Type | Processor | RAM | Display | Accessory | Video Card |
|-------------------------------------|--|-----------|---|--|---|
| Minimum (basic usage) hardware | 32-bit dual core 2Ghz CPU with SSE2 support. | 2 GB RAM | 24 bits 1280×768 display | Mouse or trackpad | OpenGL 2.1 compatible graphics with 512 MB RAM |
| Recommended hardware | 64-bit quad-core CPU | 8 GB RAM | Full HD display with 24-bit colour | Three-button mouse | OpenGL 3.2 compatible graphics with 2 GB RAM |
| Optimal (production-grade) hardware | 64-bit eight-core CPU | 16 GB RAM | Two full HD displays with 24-bit colour | Three-button mouse and graphics tablet | Dual OpenGL 3.2 compatible graphics cards with 4 GB RAM |

Table 1. 1: Blender Requirement

To work with Blender, the hardware of the machine must fulfil at least the minimum requirement of the primary usage. The better hardware machine allows performing more efficient and high efficiency. CPU (Processor) in the desktop or laptop, for blender usage, are more in rendering. Compare to GPU (Graphic Card or Video Card), it will take longer time but it is more stable and not the time limitation occurs by using Blender to Render Scene. The higher RAM (Read Only Memory) allow performing more task (Multi-tasking). Equipment or accessories are optional however a Three Button mouse plays a vital role in the blender because mainly left-click are applied, right-click is used to select, and middle mouse is controlling the viewport of the 3D scene. Graphics tablet makes it easier to do some sculpting or drawing task. For the monitor, using a higher resolution with the true tone colour display or multiple displays

will be easier to manage all the work at the same time. As for the blender, one monitor for 3D viewing, and other monitors for the UV Editor Mapping and Node Editor for Materials and Textures usage.

| Recommended Hardware | |
|----------------------------|---|
| Operating System | Windows 7/8 64-bit |
| Processor | Quad-core Intel or AMD, 2.5 GHz or faster |
| Memory | 8 GB RAM |
| Video Card/DirectX Version | DirectX 11 compatible graphics card |

Table 1. 2: Unreal Engine Requirement

Unreal Engine software is listed down in the recommended hardware. This action is to ensure the developers to operate a nice and smooth application and facing minor error. For the developers who are decided to use the Unreal Engine are required to meet the hardware requirement. For example, the OS (Operating System) for Window user must at least have Window 7 or above, with a better processor and memory, and a compatible video card to perform the work in the editor.

1.7.2 Software

Software that is using in the project is Blender, Unreal Engine, Visual Studio, Audacity, Adobe Series Software such as Photoshop, Illustrator, Fuse, AwesomeBump or CrazyBump, GitHub Client, SourceTree, and MakeHuman. More detail will be discussing in **Chapter 2.5, Development Tools Background.**

1.7.2.1 Blender

Blender is an open source 3D creation software. It supports the entirety of 3D pipelines, such as modelling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. I will use a blender to create my 3D object mesh, apply with UV Texture Map and added the rigging or armatures(Bones) for the animation as well. To Export into FBX file format and import and use in Game Engine.

1.7.2.2 Adobe Fuse and MakeHuman

Adobe Fuse and MakeHuman, are the software will be used to create hero or character of the game. It can create a temporary hero for reference. Export and import into

Blender to make a change. The purpose of this software is to create a hero for references, to improve the processing time.

1.7.2.3 Unreal Engine

Unreal Engine is a suite that integrated several tools for game developers to design and build, simulations, and visualisations for the game. Unreal Engine provided Engine Code for a developer to develop the game as what they desired. Unreal Engine or the application using is called Unreal Editor; it allows the user to place the props, coding, testing, visualise and many other features. Unreal Engine is the Triple-A game engine. The quality of this game engine is top notch and realistic. It is one of the best engines because of its excellent features for the animation and materials quality. I will use the Unreal Engine to develop the project game.

1.7.2.4 Visual Studio

Visual Studio is the IDE (integrated development environment) to provide a platform for the developer to write the code, compile and run it to get the final result. Visual Studio is connected to the Unreal Engine, because the programming language in Unreal Engine is the C++, and Visual Studio can compile C++ too. The game project will be written by the code using Visual Studio and compiled it.

1.7.2.5 Audacity

Audacity is a free source software to allow the user to record and modify the sound. This software allows me to record the sound effects or the script in the game. The software can also remove noise, to make sound sounds clearer.

1.7.2.6 Adobe Photoshop and Illustrator

Adobe Photoshop and Illustrator are used for testing and trying the created or modified of the final result of the images.

1.7.2.7 AwesomeBump and CrazyBump

Both of the software is to make the different texture map, such as normal, high map, displace map. It will test and use to get the beautiful result of any texture while applying in game.

1.7.2.8 Git, GitHub Client and SourceTree

Git is a visual control system that allows the user to upload the file and backup, it can also tracking changes in the files and coordinating work on the files among few people,

in the Git we can see the history and review it. The best things of the visual control system are we can return to any stage or commit that we want. For example, my team realised a problem on day 5, so we can return to the previous commit and test it out what is the problem occur, to reduce the code or reconstruct may take longer time analysed by human eyes. GitHub Client and SourceTree are client site software; it connects the user to clone or add the file quickly.

1.8 Project schedule / Time management

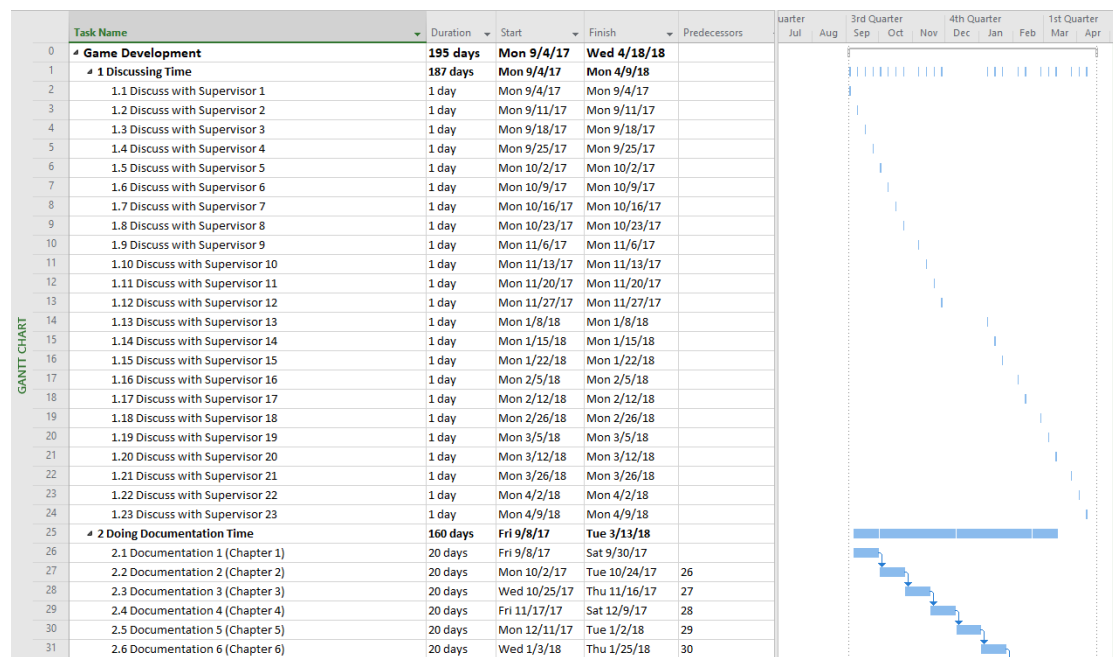


Figure 1. 1: Time Management and Gantt Chart (part 1)

| | | | | | |
|----|---|-----------------|---------------------|---------------------|----|
| 32 | 2.7 Documentation 7 (Chapter 7) | 20 days | Fri 1/26/18 | Sat 2/17/18 | 31 |
| 33 | 2.8 Documentation 8 (Chapter 8) | 20 days | Mon 2/19/18 | Tue 3/13/18 | 32 |
| 34 | ➤ 3 Planning Phase | 8 days | Mon 9/4/17 | Tue 9/12/17 | |
| 35 | 3.1 Find Sources about learning | 3 days | Mon 9/4/17 | Wed 9/6/17 | |
| 36 | 3.2 Find fundamental resources about to use | 3 days | Thu 9/7/17 | Sat 9/9/17 | 35 |
| 37 | 3.3 Final Decision about all sources | 2 days | Mon 9/11/17 | Tue 9/12/17 | 36 |
| 38 | ➤ 4 Story Phase | 107 days | Thu 11/30/17 | Tue 4/3/18 | |
| 39 | 4.1 Story Planning | 5 days | Thu 11/30/17 | Tue 12/5/17 | 46 |
| 40 | 4.2 Improve of Story | 5 days | Sat 12/23/17 | Thu 12/28/17 | 53 |
| 41 | 4.3 Better Storyline | 5 days | Thu 2/1/18 | Tue 2/6/18 | 63 |
| 42 | 4.4 Extend Story | 5 days | Thu 3/29/18 | Tue 4/3/18 | 76 |
| 43 | ➤ 5 Learning Phase | 67 days | Wed 9/13/17 | Wed 11/29/17 | |
| 44 | 5.1 Learning Fundamental Programming | 7 days | Wed 9/13/17 | Wed 9/20/17 | 37 |
| 45 | 5.2 Learning Unreal Fundamental | 30 days | Thu 9/21/17 | Wed 10/25/17 | 44 |
| 46 | 5.3 Learning Blender Fundamental | 30 days | Thu 10/26/17 | Wed 11/29/17 | 45 |
| 47 | ➤ 6 Simple Developing Phase | 20 days | Thu 11/30/17 | Fri 12/22/17 | |
| 48 | 6.1 Create Basic Model | 4 days | Thu 11/30/17 | Mon 12/4/17 | 46 |
| 49 | 6.2 Apply in Game | 1 day | Tue 12/5/17 | Tue 12/5/17 | 48 |
| 50 | 6.3 Develop simple Player Controller | 4 days | Wed 12/6/17 | Sat 12/9/17 | 49 |
| 51 | 6.4 Develop simple AI Controller | 4 days | Mon 12/11/17 | Thu 12/14/17 | 50 |
| 52 | 6.5 Explore checkpoint | 4 days | Fri 12/15/17 | Tue 12/19/17 | 51 |
| 53 | 6.6 Improve UI | 3 days | Wed 12/20/17 | Fri 12/22/17 | 52 |
| 54 | ➤ 7 Intermediate Developing Phase | 34 days | Sat 12/23/17 | Wed 1/31/18 | |
| 55 | 7.1 Create 3D model with mesh | 5 days | Sat 12/23/17 | Thu 12/28/17 | 53 |
| 56 | 7.2 Environment Animation | 5 days | Fri 12/29/17 | Wed 1/3/18 | 55 |
| 57 | 7.3 Implement In Object In Game | 2 days | Thu 1/4/18 | Fri 1/5/18 | 56 |
| 58 | 7.4 Improve Player Controller | 5 days | Sat 1/6/18 | Thu 1/11/18 | 57 |
| 59 | 7.5 Improve AI Controller | 5 days | Fri 1/12/18 | Wed 1/17/18 | 58 |
| 60 | 7.6 Improve In Game Animation | 4 days | Thu 1/18/18 | Mon 1/22/18 | 59 |
| 61 | 7.7 Feature improve such as checkpoint | 4 days | Tue 1/23/18 | Fri 1/26/18 | 60 |

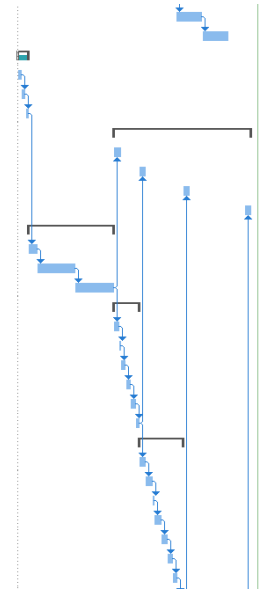


Figure 1. 2: Time Management and Gantt Chart (part 2)

| | | | | | |
|----|--|----------------|--------------------|--------------------|----|
| 62 | 7.8 Better UI | 2 days | Sat 1/27/18 | Mon 1/29/18 | 61 |
| 63 | 7.9 Recorrect all the object position | 2 days | Tue 1/30/18 | Wed 1/31/18 | 62 |
| 64 | ➤ 8 Advance Developing Phase | 34 days | Thu 2/1/18 | Mon 3/12/18 | |
| 65 | 8.1 Better and More 3D model | 7 days | Thu 2/1/18 | Thu 2/8/18 | 63 |
| 66 | 8.2 Better Animation in game | 7 days | Fri 2/9/18 | Fri 2/16/18 | 65 |
| 67 | 8.3 Place Object in Game | 4 days | Sat 2/17/18 | Wed 2/21/18 | 66 |
| 68 | 8.4 Better Player Controller | 7 days | Thu 2/22/18 | Thu 3/1/18 | 67 |
| 69 | 8.5 Better AI Controller | 7 days | Fri 3/2/18 | Fri 3/9/18 | 68 |
| 70 | 8.6 In deep checking and testing | 2 days | Sat 3/10/18 | Mon 3/12/18 | 69 |
| 71 | ➤ 9 Marketing Phase | 14 days | Tue 3/13/18 | Wed 3/28/18 | |
| 72 | 9.1 Develop Official Web | 7 days | Tue 3/13/18 | Tue 3/20/18 | 70 |
| 73 | 9.2 Promote Certain platform free to test | 1 day | Wed 3/21/18 | Wed 3/21/18 | 72 |
| 74 | 9.3 Requiring funding from certain company | 3 days | Thu 3/22/18 | Sat 3/24/18 | 73 |
| 75 | 9.4 Upload product to get Fund | 2 days | Mon 3/26/18 | Tue 3/27/18 | 74 |
| 76 | 9.5 Advertisement | 1 day | Wed 3/28/18 | Wed 3/28/18 | 75 |
| 77 | ➤ 10 Next Level Phase | 17 days | Thu 3/29/18 | Tue 4/17/18 | |
| 78 | 10.1 Require selling product on STEAM platform | 2 days | Thu 3/29/18 | Fri 3/30/18 | 76 |
| 79 | 10.2 Update in game graphic | 4 days | Sat 3/31/18 | Wed 4/4/18 | 78 |
| 80 | 10.3 Update Game feature | 11 days | Thu 4/5/18 | Tue 4/17/18 | 79 |
| 81 | ➤ 11 Final Task | 1 day | Wed 4/18/18 | Wed 4/18/18 | |
| 82 | 11.1 Presentation | 1 day | Wed 4/18/18 | Wed 4/18/18 | |

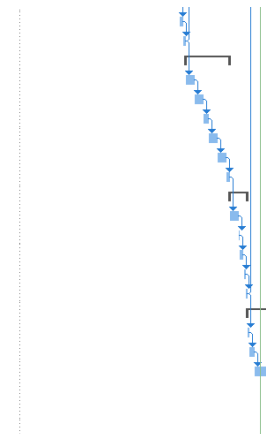


Figure 1. 3: Time Management and Gantt Chart (part 3)

1.9 Summary questionnaire result

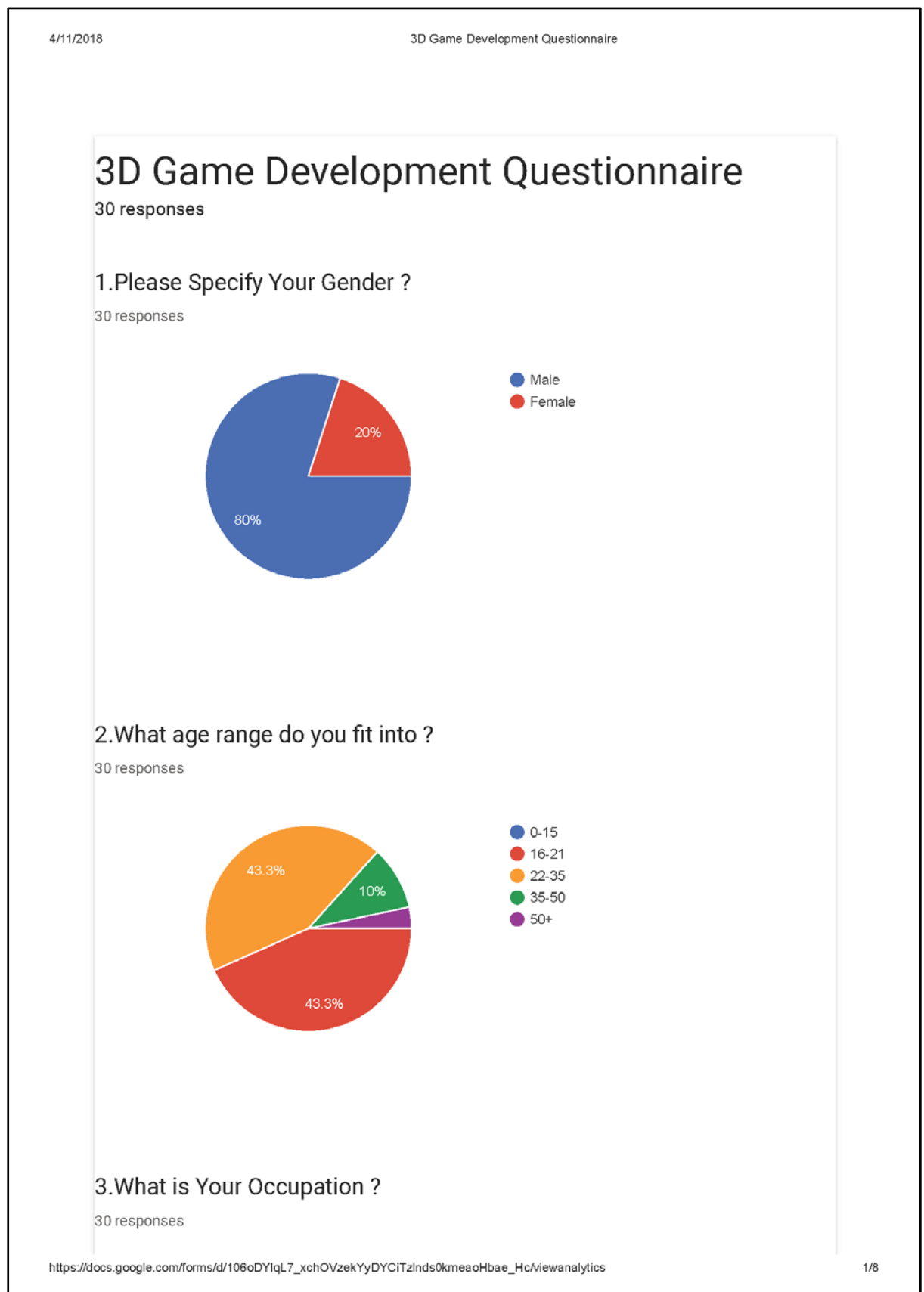


Figure 1. 4: 3D Game Development Questionnaire Result (part 1)

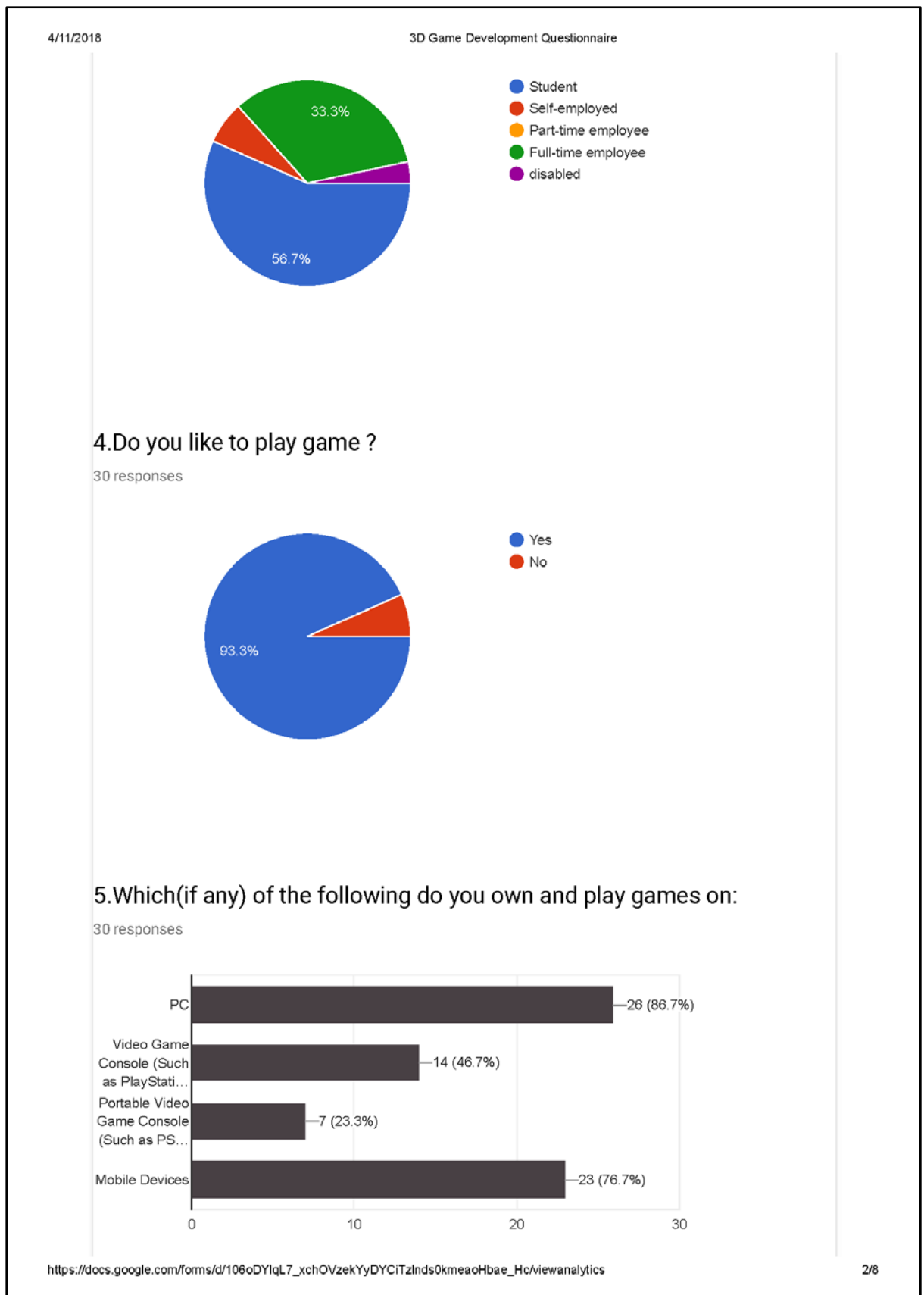


Figure 1. 5: 3D Game Development Questionnaire Result (part 2)

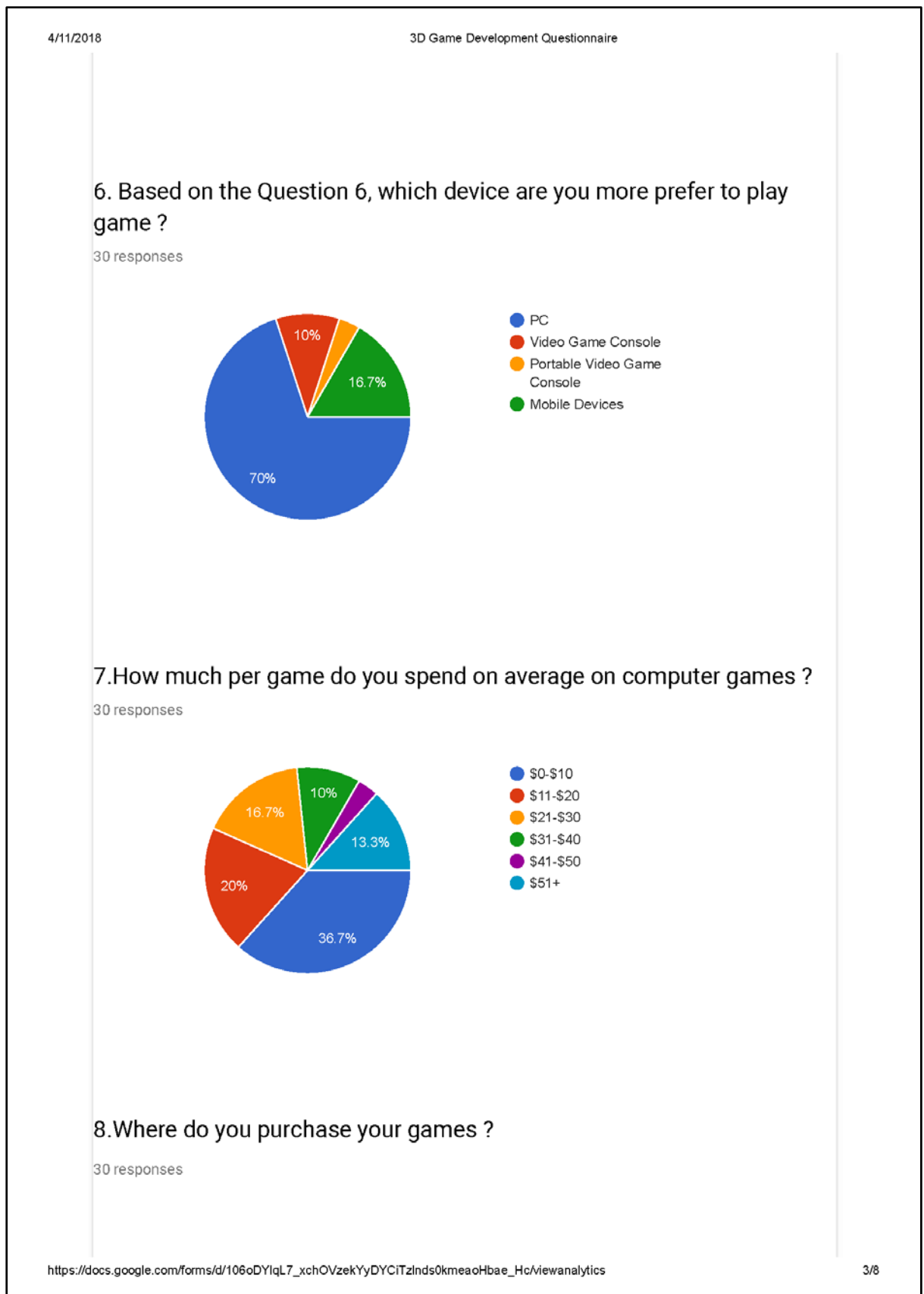


Figure 1. 6: 3D Game Development Questionnaire Result (part 3)

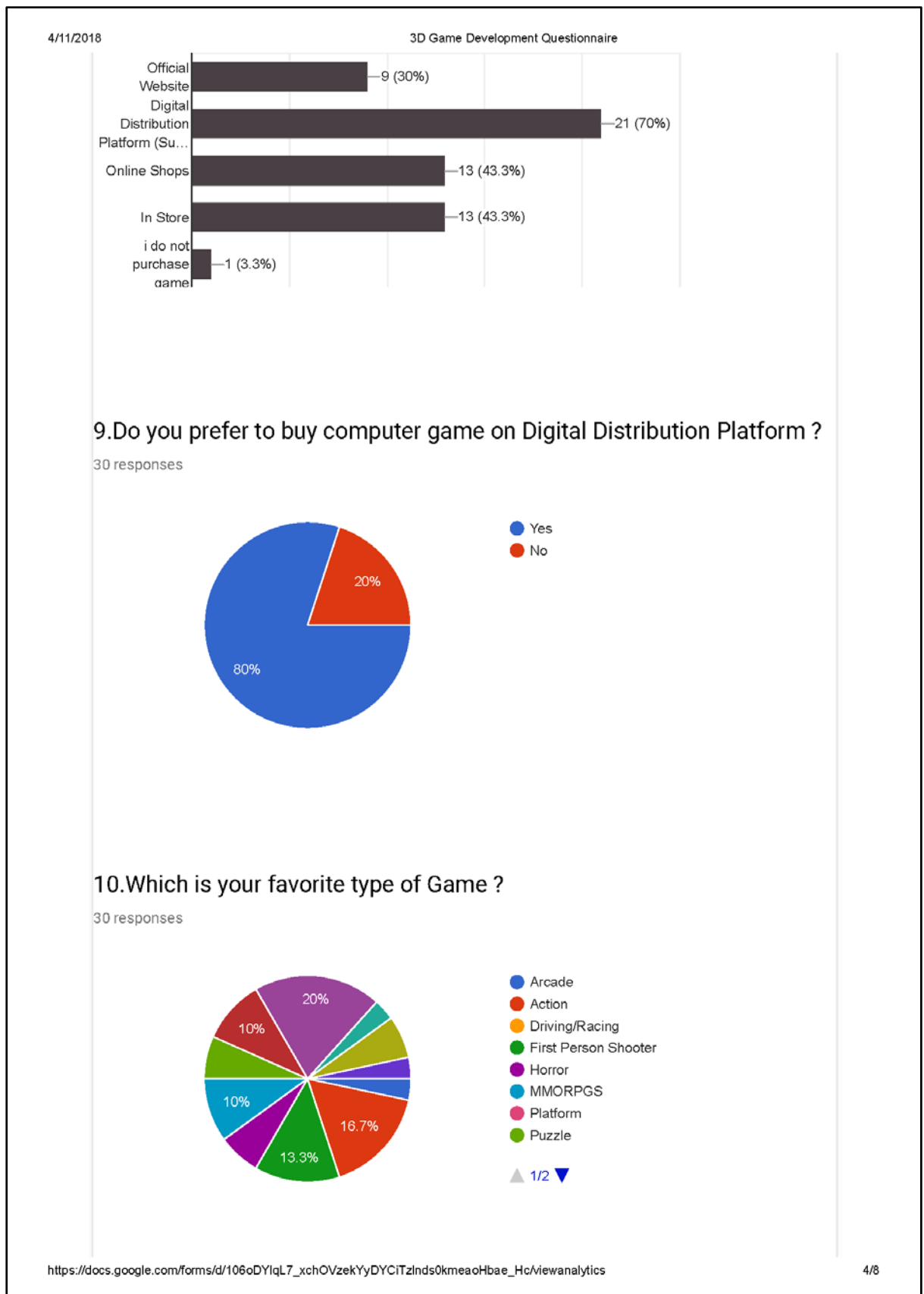


Figure 1. 7: 3D Game Development Questionnaire Result (part 4)

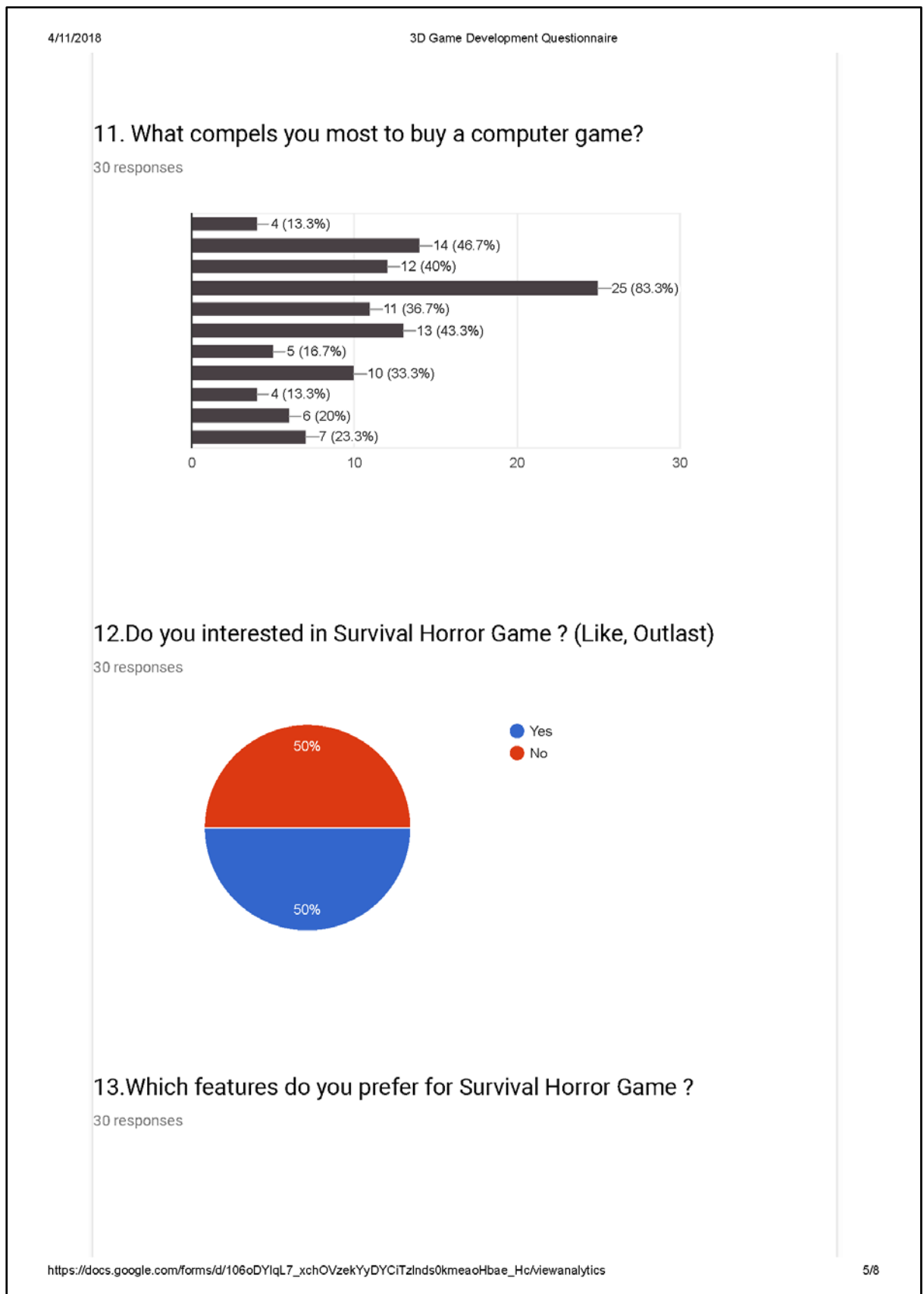


Figure 1. 8: 3D Game Development Questionnaire Result (part 5)

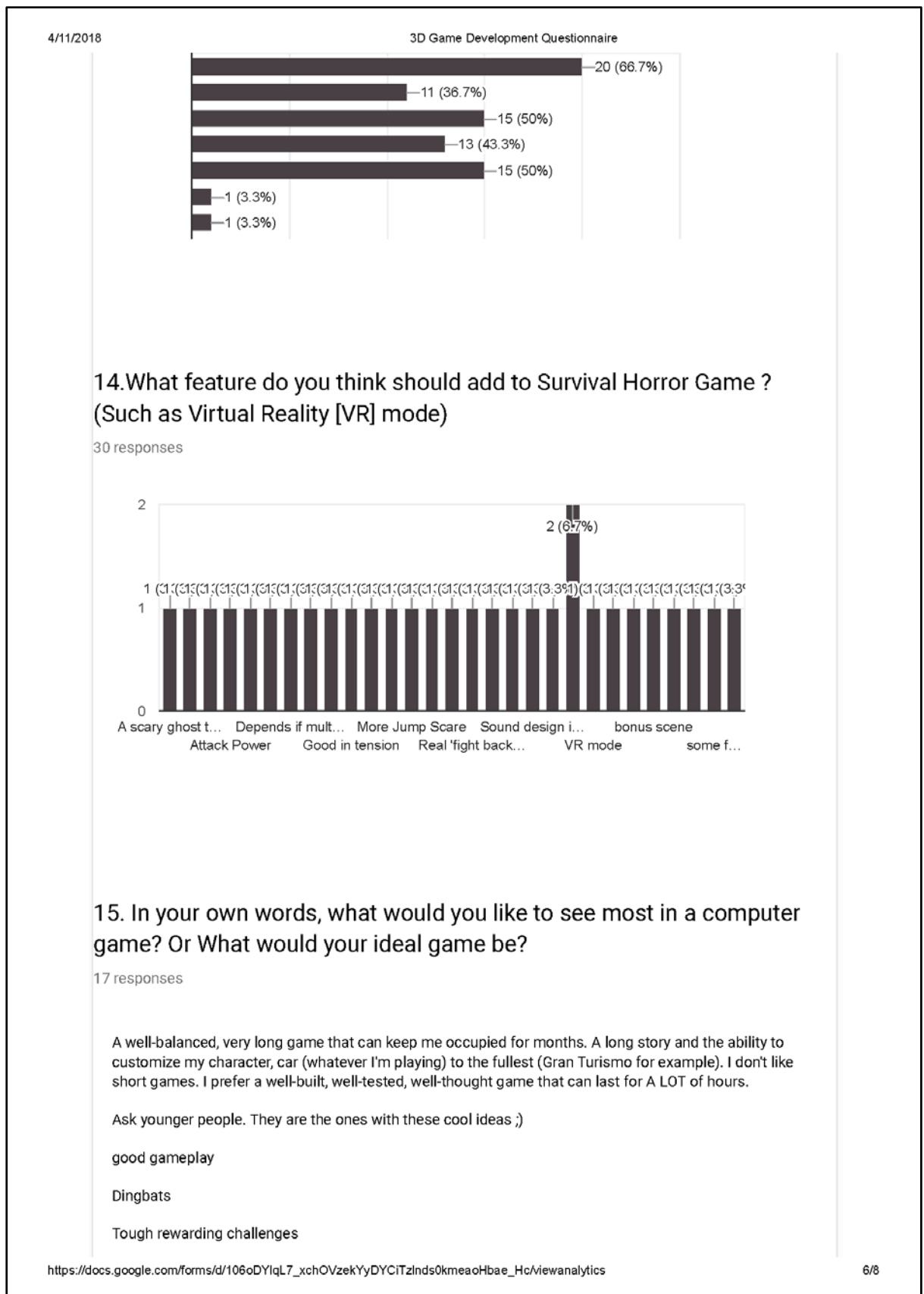


Figure 1. 9: 3D Game Development Questionnaire Result (part 6)

4/11/2018

3D Game Development Questionnaire

A game with a good narrative (like Horizon ZD, or Witcher 3), good graphics, a loot of possibilities (like Skyrim). Also the OST and translate the game to other langages is very important.

More action, less side missions.

-

The computer games have a beneficial picture.

Destiny 2

A good story

Moral choices with powerful consequences.

Detailed Character Customization

Deuteros + Populous + Flight Sim + Kerbal Space Program + GTA

storyline and immersive flow

A game where many players can play together, where we cooperate and arrange community events, where there is no pay to win and where flamers and hackers get banned in order to keep them away from the normal members of community.

Open World, multiple endings.

Last Question, the answer can allow us to track your record, will enable us to send an updating news to you, and you will be chosen as one of the lucky fellows to receive the game when had been developed and published

Email Address (Optional)

11 responses

brian.grubba@gmail.com

biscuitzmonster@gmail.com

miguelexposito3d@gmail.com

inzamam99@gmail.com

michaelgnolan@hotmail.com

jackdraak@gmail.com

yee.joeqin@ypccollege.edu.my

ruimingrm96@gmail.com

https://docs.google.com/forms/d/106oDYlqL7_xchOVzekYyDYCiTzInds0kmeaoHbae_Hc/viewanalytics

7/8

Figure 1. 10: 3D Game Development Questionnaire Result (part 7)

4/11/2018 3D Game Development Questionnaire

dar_mcbain@yahoo.co.uk
oskar4485@gmail.com
hamge96@hotmail.com

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms

https://docs.google.com/forms/d/106oDYlqL7_xchOVzekYyDYCiTzInds0kmeaoHbae_Hc/viewanalytics 8/8

Figure 1. 11: 3D Game Development Questionnaire Result (part 8)

1.10 Conclusion

In this section is the summary of chapter 1. It is about the video games is one of the most critical entertainments in the current generation, everyone will take a short break to play the game to relax. As a horror game developer of the project, it is necessary to bring out the scary feeling, and several emotional inside the game when playing. This game will try to pick up several parts that are quite memorable from the movie and implement in the game to make it more realistic. Nevertheless, it may have the possibility to alter the trend after the game was released. Hardware and software are required to meet the requirement to allow the developer to build or develop in a seamless workplace, and resulting in a high graphics game. To make the game famous, and let a minority of the audience to step out from the minority group, encourage to change themselves and solve their problem.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Computer games and video games have turned into exact prevalent in adolescents' lives and play a prominent role in the culture of young people (Dorman, 1997). Games can play everywhere in technology-rich environments equipped with laptops, smartphones, game consoles (mobile and stationary), set-top, boxes and other digital devices. From this phenomenon, is believed that the intrinsic motivation young people show towards games be able to combine with educational content and objectives into "digital game-based learning" (Prensky, 2003). The idea of using games to engage audient in the progression of active learning is not fresh. Over several years ago, Developers have been gradually more incorporating various games into the specific curriculum to create fun and engaging learning environment for the player. Even though this can be situated challenging and time-consuming, is challenging to ensure the player is fully participating in playing and while learning. Video games can make a positive, fun, entertaining atmosphere amongst players (Wolf, 2001). The first feature of a video game that differentiates them from other media is their need for player interaction. To discover and advance in the game world, the player has to have a certain amount of skill (Wolf, 2001). The reason why the game can be used to part of the learning materials, Games are played to win or achieve a goal. The key to motivation is winning while remaining challenged. They motivate via fun, 'part of the natural learning process in human development' (Bisson C, 1996) and instant, visual feedback (Prensky, 2001). This is correct both of 'mini-games', where players accomplish quick results (Mitchell, 2003) , and of complex games, such as fantasy or simulation games, which have goals and sub-goals.

This Chapter discuss literature review about certain of terminology or keyword is related to this project development. Literature reviews are collected from several sources such as from books, journals, articles, websites and existing research. These collections are used for guides and explain understanding Video Game Development.

Correlation between previous research and this study will be highlighted in this chapter. Few of other similar game or application will be compared. Methodologies, which also known as SLDC (Software Development Life Cycles) is a guide to the development process for the determined direction of development software or system. The discussion of methodologies will be included in chapter along with detail explanation, the brief of significant the advantages and disadvantages regarding the selected methodologies that deemed suitable for the project to carry out. The final decision of the methodology to be used will be more discussion in the next chapter, Chapter 4.

As video games have come to be more widely used by different ages, it has become easier to implement them as an educational tool, which can positively affect the player's knowledge and skills if applied efficiently (Bellotti, et al., 2009). Educational games can be developed using a state-of-the-art commercial game development approach with the enhancement of developed educational models (Bellotti, et al., 2009). The game project will be part of the educational game cause have been involving certain level to let the player figure out how to solve and escape the place. The player will be improving the skill and learning while playing or after completing the game.

Video games are considered a useful tool within the education context because of their positive impact on players' cognition and knowledge (Chuang & Wei-Fan, 2007) (Aguilera & Mendiz, 2003).

These positive impacts include:

2.1.1 Improving cognitive and reading skills.

Video Game can be improving the reading skills because users are excited and enjoying the game, treat the user own like one of the players in real life. Due to the video game of the developer is assurance the player understands the game of the situation, story, criteria. The HUD (head up display) or UI (User Interface) will be showing some

instructions, quest, mission. It could be just an only text description or may involve with the image. The player will be learning the new word while enjoying the game.

2.1.2 Motivating logical thinking process.

Some of the video game are involved few of the education situation. The genre of Strategy, the game will be involving such problem-solving quest or mission in the game, like Puzzle type. Player need to complete the game is needed to solving the question to pass by the obstacle.

2.1.3 Strengthening observational skills.

Every game is now a day, will improve the observation skill of the player. This is because the user is entirely focusing and enjoying the video game, the player will keep an eagle eye on the environment in the game, understand what situation are, and trying to win the game.

2.1.4 Acquiring basic and factual knowledge.

Some game will be using the actual history to become the game background. The player will be understanding the case of history while playing the game, and may find in deep if interested and want to know more. The same item will be used as a general item in real life, however not majority people understand how to use. In game may showing the simple step and possible to gain some knowledge about what is the item and how to use for.

2.1.5 Enhancing the abilities of problem-solving and decision-making.

Mostly the player can be acquiring problem-solving in the game are those of game will have several ending scenes. The game of the flow how to running out is decided by the user itself. Each time of the decision will change the path of the game line. The game will be more fun and realistic. The player is based on the particular question and decision to enhancing the abilities of problem-solving in a particular situation. The game such as Until Dawn, The Witcher, Minecraft Story mode.

2.1.6 Developing strategic planning.

Based on the previous point have been a mention about that how the user can be learning and improve every specific skill. From all these skills will be gathering all the information and developing the own strategic or tactile in game. The player can quickly to planning a plan or strategy. This improvement will be useful in real life.

2.1.7 Supporting spatial awareness.

Each video game is different genre or may involve two or more. The player can be improving the spatial awareness and support in real life. Such as sport, music.

2.2 Research on the existing system/web domain (or previous system)

In the earlier twenty years, video game industry has established significant contributor to global entertainment economy (Marchand & Hennig-Thurau, 2013). Jane McGonigal had been mention before the four attribute (McGonigal, 2011)

2.2.1 Goal

The goal is the specific result; players will work to accomplish. It focuses their attention and repeatedly turns their participation throughout the game. The goal provides players with a sense of purpose.

2.2.2 Rules

The rules place limitations on how players can achieve the goal. By eliminating or limiting the noticeable ways of getting to the goal, the rules push players to explore previously uncharted possibility spaces. They release creativity and foster strategic thinking.

2.2.3 Feedback

The feedback system tells players how close they are to succeeding the goal. It can take the form of points, levels, a score, or a progress bar. Alternatively, in its most basic form, it can be as simple as the player's knowledge of an objective outcome: "The game is over when..." Real-time feedback serves as an ability to the players that the goal is achievable, and it motivates to keep playing.

2.2.4 Participation

Voluntary participation requires that each person who is playing the game knowledgeably and willingly accept the goal, the rules, and the feedback. Knowingness establishes common ground for multiple people to play together. Moreover, the freedom to enter or leave a game at will ensures that intentionally stressful and challenging work is experienced as the safe and pleasurable activity.

2.2.5 Slender: The Arrival



Figure 2. 1: Slender: The Arrival of Steam (Midnight City, Majesco Entertainment, 2013)

Based on Figure 2.1. Slender: The Arrival is a single player indie horror game. This game is the official video game alteration of Slender Man, developed in collaboration with Eric "Victor Surge" Knudson, creator of the paranormal phenomenon has been frightening the curious-minded around the world since its initiation, with Mark Hadley and Blue Isle Studios. Blue Isle Studios is an independent game design company based out of Toronto, Canada. Due to the last series of the game was short (Slender: The Eight Pages). This investigational game helps out to breathe new life into the horror genre and captivated gamers around the world through its use of pure, unadulterated fear. Slender: The Arrival is re-imagining an extension of original game created by Mark Hadley, teamed up with writers behind Marble Hornets series and development team at Blue Isle Studios. The Arrival features is a brand new storyline, better-quality visuals, great replay value, and most importantly, survival horror at its best. The slogan

of the game is “You are on your own. No one to come for you. No one to help you. No one to hear you scream.”



Figure 2. 2: Comment for Slender: The Arrival of Steam (Comment, 2013)

Based on figure 2.2. The Positive comment from Britt_Tenille user, He recommends the game to another user because the game is fun and entertaining story. However, the guidance of the game is not doing well for any novice user. Next, the Negative comment from the user Mr.Muffs, He mentions that the game is mostly just walking around the area and try to see through the dark place. The same case as the Positive Comment of

the first user, He also gets lost in the game. Last, the game is not really to make it emotional to the user Mr Muffs.

2.2.6 Outlast



Figure 2. 3: Outlast from Steam (Red Barrels, 2013)

Based on Figure 2.1. Outlast is a single player survivor horror indie game. A first-person survival horror game developed by veterans of some of the biggest game franchises in history. The player is as an investigative journalist Miles Upshur, due to the job requirements, explore Mount Massive Asylum and try to survive long to discover the terrible secret. The Synopsis of this game is in Colorado, a place of remote mountains. Inside the Mount Massive Asylum, horror is waiting. A branch of “research and charity” of Murkoff Corporation have re-opened a long-abandoned home for mentally ill, but the actual asylum has been operating in strict secrecy. The journalist as a player has received a tip from an anonymous source. He breaks into the facility, and what he discovers walks a terrifying line between religion and science, nature and else entirely. When he inside the place, his only hope to escape the lies with the terrible truth at heart of Mount Massive. Outlast is real survival horror experience, which intentions to a demonstration that scary monsters of all derived from the human mind.

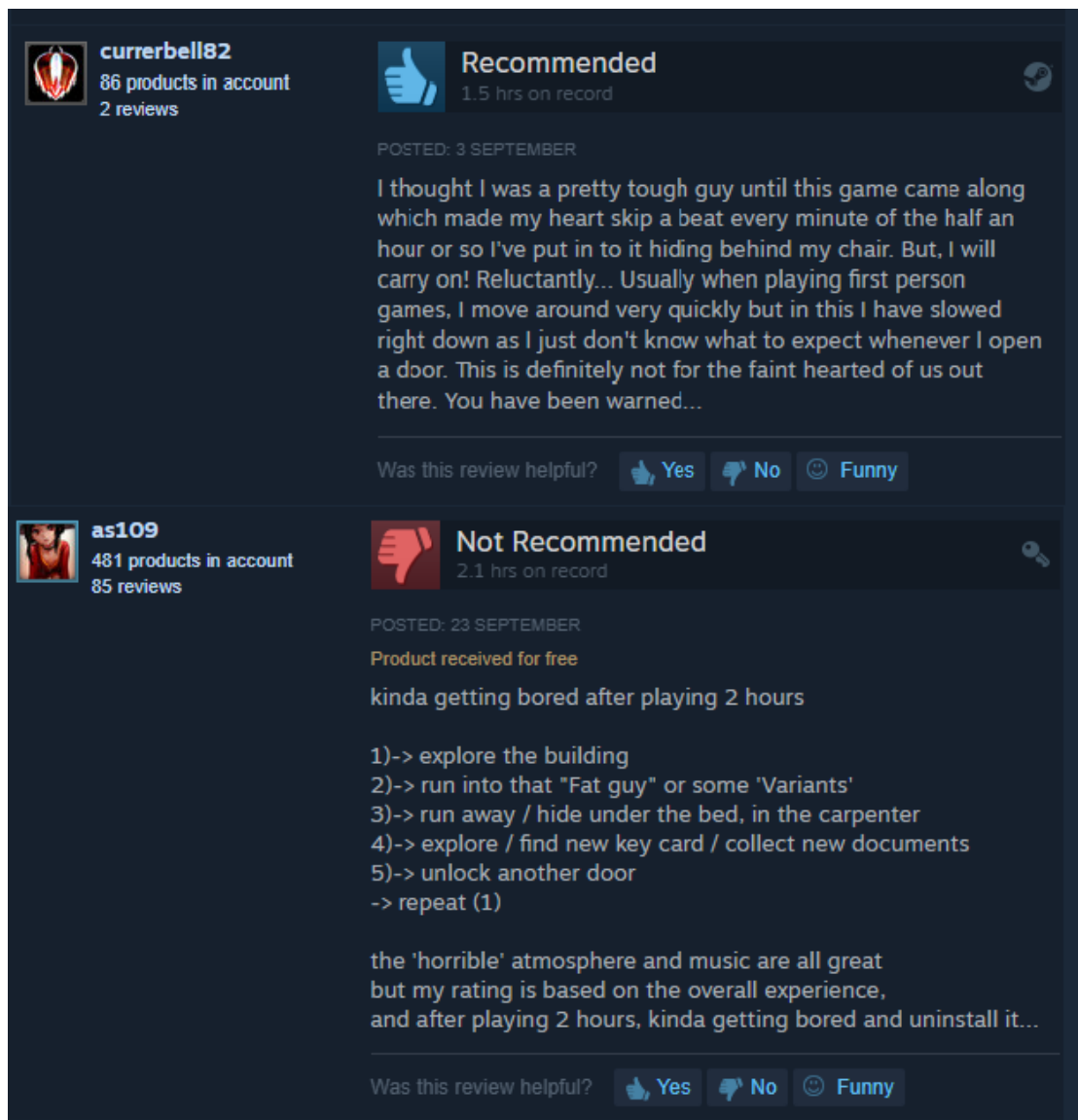


Figure 2. 4: Comment for Outlast from Steam (Comment, 2013)

Based on figure 2.4. Some user will recommend the game because the game is scary much. From the user currebell82 mention on the comment. He said, he first thought he was a terrible guy until playing the game, he realises the game can have made his heartbeat is skipped every minute of the time while he was playing. However, some particular user will be getting bored of the game. Based on the user as109. The Player mention that the reason due to the game is to keep repeating the processing. On the other hand, meant that the game keeps playing the same things.

2.2.7 Emily Wants To Play



Figure 2. 5: Emily Wants To Play from Steam (SKH Apps, 2015)

From the figure 2.3, Emily Wants to Play is a survival horror game. The game is set in a creepy house with a girl called Emily and her dolls; Player plays as a pizza deliverer that is locked in and must survive in the night, playing Emily's games. What Emily Wants to Play is available for significant platforms such as PC, Mac, PlayStation 4, Xbox One, iOS and Android. The game works well with HTC Vive and Oculus Rift on the PC version. The synopsis of the game is played at night of 11 pm. Player to delivering the pizza, the house is the last order on the route. The house of the lights is on, and the front door is opened. However, the windows are all cover up, and the grass is overgrown. The area seems like the strange place to deliver pizza, on the other hand, looks like someone at home. The character is running into the house because the storming is terrible. The player is announcing, however, the place is silence. The rain is enormous; the character only can step inside the door and look around. However, the front door unexpectedly closes. The player is just wanted to finish the delivery and back home for the night. However, now need to figure out how to exit this creepy house. Freely to roam in the house and try to figure out what that happened. The girl (Emily) and the dolls will start roaming the house as the hour's toll by. Stay away and keep safe, however, if do end up in a room with other, figure out how to stay alive.



Figure 2. 6: Comment for Emily Wants To Play from Steam (Comment, 2015)

Based on the comment, user Jessifire is satisfied the game however it seen the game have some bug, so will make the user more frustrating and exit the game. Another user review is negative, Shadow_Wolf mentions that the game is not scary and a short story of the game. However, the user also introduces the game for the beginner who's are want to start they own journal for the horror game.

Comparison Existing Game

| Game Name | Features | | | | |
|----------------------|--------------------|-------------------|--------------------|-----------------|------------------|
| | Multi-Story Ending | Random Jump Scare | Immersive Graphics | Limited Sources | Player Hide able |
| Slender: The Arrival | Yes | Yes | Yes | Yes | No |
| Outlast | No | No | Yes | Yes | Yes |
| Emily Wants To Play | Yes | Yes | No | No | No |
| You Are Not Alone | Yes | Yes | Yes | Yes | Yes |

Table 2. 1: Comparison of Existing Game

Based on Table 2.1, We can understand the difference between these of the game. Each game has supported the own features and some may not too. The game of our project (You Are Not Alone), is inspired by the game Outlast. Outlast is the first person horror game. That is the indie game. However, the game is getting a mostly positive review from every different website, platform. Outlast is the excellent game. However, some feature is not included. Outlast have an immersive graphic; it can be said like AAA (Triple-A) game. Now the day of the audient will more focus on the graphics first before the storyline, gameplay. The reason why Outlast will be the excellent horror game because the resources in game to use are less or limited. So that, the player needs to manage and control every single time when deciding to use the resources. However, the game does not have the option to make the story path change, the story of the game does not have a multi-story ending. Is meaning that, the game only has one ending line. The game is scary however don't have the random jump scare, just only a particular monster or human to attack and chasing the player. In the end, the particular game does not allow player hide in the game. Mean that the user does not allow to hide from avoiding user chasing. Hiding is the only hope in the game to prevent enemy keep chasing the player, and can take a breath.

2.3 Findings on the existing system/web domain (or previous system)

Based on the section 2.2 Research on the Existing System (Previous System). The Project can be concluded; the game is inspiring from the Outlast. Outlast is a great game even after a few years now day, audient is like to play and recommend it. The game is having some lack need to improve so that the reason why You Are Not Alone will be created. Outlast have the immersive graphics or the term call as Triple-A game (AAA), because the graphic is nice, most of the audient are attractive. Outlast is the first person survivor horror game, it involving the immersive graphic, the first impression of the game is best. The gameplay of the game is excellent, playing like hiding and seek and solving puzzle game to find a way to escape the place. Because the game is the best to review, one of the reasons of that, since the game is less or limited resource use in the game. The game is making worrying about the player; each time the resources are almost finished or gone.

Our Project, You Are Not Alone. Will follow the path and trying to make the game are better even some feature it looks like similar, but the enjoyment of the game is a different feeling. You Are Not Alone will be similar to Outlast, the game it will be same as first person survivor horror game. The player is not allowing to do any attacking, just can hide. Our game will be trying the best to improve the graphic to become more realistic compared to Outlast. The project of us will have limited resource use; this can make a terrifying of the player play while resources are finished, such as a battery of the flashlight. Outlast are don't have any multi-ending, You Are Not Alone will make it the game have several options for ending scene. The path of ending will depend on how the player plays style such as collecting an item and figure out the secret puzzle question. Outlast will have a human or monster to chase and attack the player, however, are not or less the jump scare. The project of us will be improved and making the game scarier. Random Jump Scare is a way to make the player tension in the game. The random jump scare will have several different versions. Such as only sounds (like footstep on specific place, or laughing sound in specific location), Spawning the actor with or without the sound effect and staring the player in few second after that disappear, and last, will be spawning the actor behind the player and start to chasing the player until specific time than disappear. It may high chance to get some jump scare during light off. The last feature it will be improved, that is Player Able to Hide in the game. When the User was kept chasing the Enemy. It should need a place for the player take a break. The reason why is because the user will have limited stamina, however, if the Enemy is non-stop to chasing the player, more of the time will be complicated to survive in the game. The feature will make the game more reliable and realistic. In short, You Are Not Alone being inspiring from the game called Outlast and will be following some specific path and make a change to improve the game to get the better result.

2.4 Development methodology background

A project is a success and managed well, which is a manager or team to carry out a software development method or methodology to manage a project efficiently. Software or System Development Methodology also as Software Development Life Cycle (SDLC). SDLC is called as Software Development Process. Software industry to plan, create (design, develop) and test the high-quality software by using this process. The reason why SDLC are born for because to aims for producing high-quality software

or system and meets the customer expectation completed the project at times and cost estimates. Possible to exceed what customer wants to. That is a structure to outline the tasks performed in every step in the process.

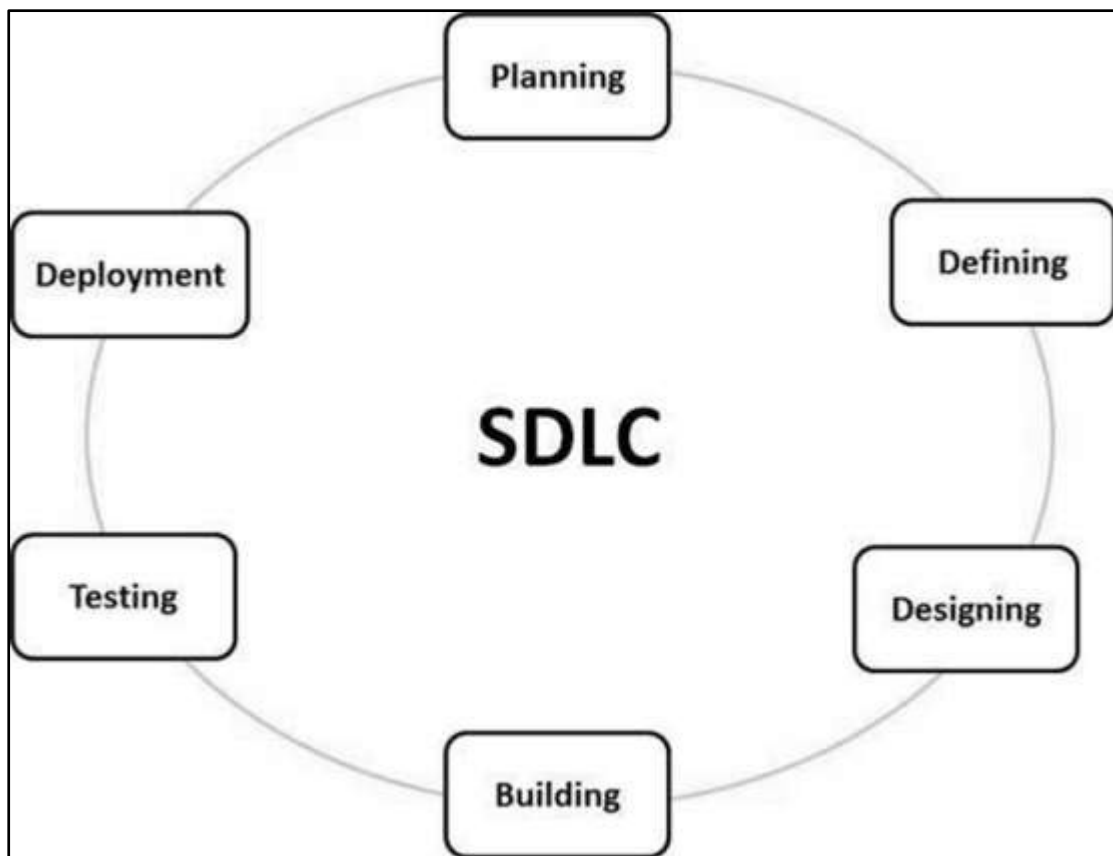


Figure 2. 7: Typical Software Development Life Cycle (SDLC, n.d.)

Based on figure 2.7, that is a typical software development lifecycle. It meant that all the development methodology would have been involved.

In the first stage, Planning and Requirement Analysis. In methodology, the most important and the primary stage is requirement analysis. The result is performed by team members with the input from the consumers, specific department (sales department), surveys and the domain industry. Those of the information will be used to plan the pre-project approach. The Planning stage, it also for assurance the quality requirements and identification of risk. The outcome of this stage is to define various way or approaches to use and follow. To make the project success with fewer risks.

The following stage is Defining Requirements. The main point on this stage is precise to define and documented product requirement and approved by customer or market analysts.

Next stage, Designing the Product Architecture. Based on the previous stage, SRS (Software Requirement Specification) reference for product architects to get with the best result for developed. The requirement specified in SRS, generally more than one design approach, it for the documented in a DDS (Design Document Specification) and product architecture proposed. DDS is selected best approach for the product by revised by all essential stakeholders and based on various parameters as risk assessment, design modularity, product robustness, time and budget constraints.

In the fourth stage, Building or Developing the Product. The actual development begins, and product to build starts from this stage of SDLC. The programming code is selected and generated as per DDS and during this stage. If the previous stage, design approach is in detailed and organised. The code can be accomplished with less hassle. The coding guidelines are necessary for developers to follow. The programming tools such as compilers, interpreters, debuggers are used to generate the code. The programming language such as C++, Java, PHP is chosen with respect the type of software being developed.

Moreover, the Testing the Product stage. Is a subset all the stage as modern SDLC models. The activities of testing are commonly involved in all the SDLC stage. Nevertheless, this stage is focused on testing only, where the product or system defects are reported, tracked, fixed, retested. Until the product reaches the standard quality as defined in SRS.

In the last stage, Deployment in the Market and Maintenance. As soon as the product is tested and prepare to deploy, released formally inappropriate market. A good business strategy of the organisation is first released in the limited segment and tested the market environment. It called UAT (User acceptance testing). Based on the feedback and comment, the product will be released an updated version of the suggested enhancements in targeting market segment. Maintenance will frequently be checking the existing customer base.

SDLC have various models defined, designed which to follow during software development process. Each model will follow series of steps, ensure success in the process of software development. The famous and model will be discussing here are Waterfall Model, Spiral Model, and V-Mode. All software development methodologies exist are used for different reasons.

2.4.1 Waterfall Model

A first process model to be introduced is the Waterfall Model. This Model is the earliest Software Development Life Cycle approach to use for software development. Waterfall Model is referred linear-sequential life cycle model. This model illustrates the linear sequential flow type of this method makes easy to understand and manage. Each phase must be 100% completed before moving to next phase can begin, and there is no overlapping in the phases.

Waterfall Model is suitable for use in a short project; requirements are well documented, definite and fixed. No any ambiguous requirements. The definition of the product is stable, and technology is not dynamic. Available plenty resources with the required expertise to support.

2.4.1.1 Waterfall Model Design

Waterfall Model approach, the process will have divided into separate phases, the result of one phase acts as input for next phase sequentially. Next phase only started after defined the goals was archived from the previous phase.

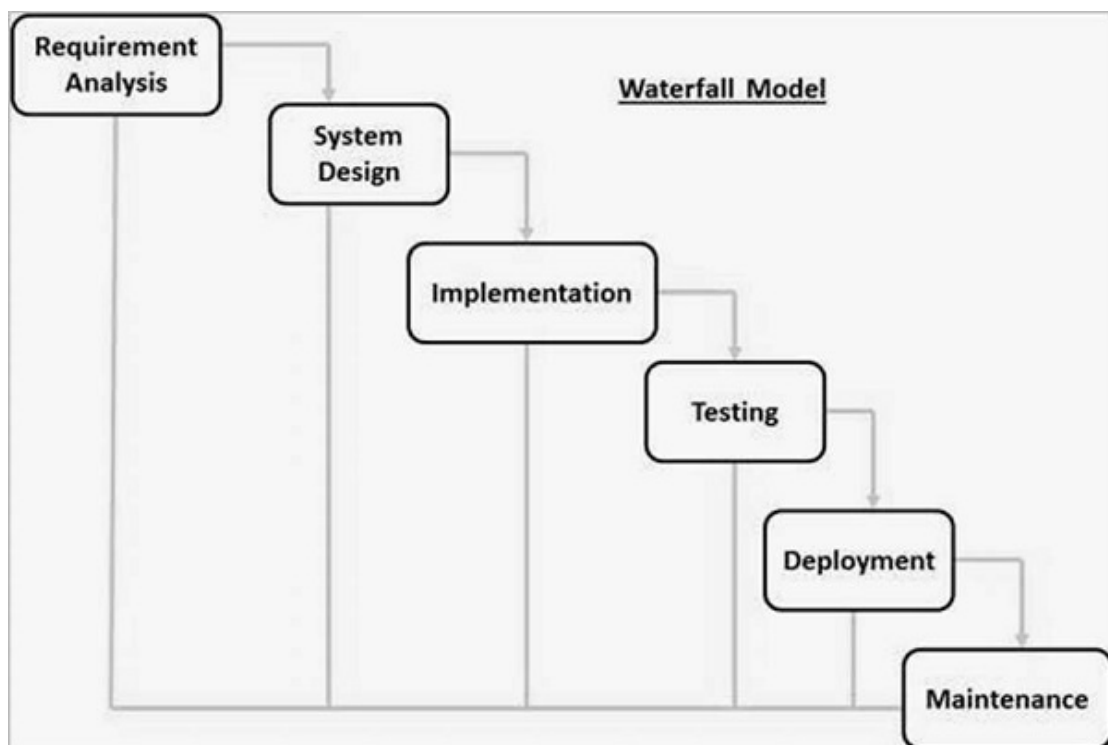


Figure 2. 8: Waterfall Model (Waterfall, n.d.)

Based on figure 2.8, that is a representation of the different phase of the Waterfall Model. The sequential phases are

2.4.1.1.1 Requirement Gathering and analysis

In this phase, all the possible requirement of the software or system to developed are captured and documented requirement specification. Example, discussing with the client, to check and confirm the product specifications.

2.4.1.1.2 System Design

In this phase, are studying the information of requirement specification from the first phase, and the design of the system is prepared. This phase, the system design helps the developer or team to specifying hardware, software and other system requirement and help in define overall system architecture. Example, based on the previous stage, translating the documentation to build a mock-up, this action is enabled to help the developer for review before developing the process.

2.4.1.1.3 Implementation

The implementation phase will get the input from the system design; the first developed system will be in small programs as units. These units will be integrated into next phase. Each unit developed and tested for functionality, which referred to Unit Testing. Example, the project will have divided into several small modules, build and test it individual before the move to the next phase of the integration process.

2.4.1.1.4 Integration and Testing

In this phase, will use the all units from previous phase (Implementation), are integrated into a system after test each unit. Post-integration entire system to test for any failures and faults. Example, from the previous stage, all of the module will be integrated into phase and testing the project when the module became one have occurred any new error.

2.4.1.1.5 Deployment of system

Once the Integrated system was tested, functional and non-functional. The product is deployed or released into the market. Example, when the project is integrated successes, the project can build and published in several marketplaces, such as Digital Distribution Platform.

2.4.1.1.6 Maintenance

The maintenance phase is to deliver the changes in the customer environment. There may have some issues which come to the client environment. To fix and solve the problem will be releasing new up-to-date patches. The updated version also will enhance the product to be a better version. Example, When the product has been deployed, user or client may find some problem in the product, the developer will solve the trouble and update the new patch into the same marketplace to allow the client is enabled to keep the system up-to-date.

2.4.1.2 Waterfall Model Advantages and Disadvantages

The brief of the pros is each stage can be set with deadlines and allowing departmentalisation and control. The brief of the cons is challenging to return and change anything at each stage. The significant advantages and disadvantages of waterfall model is

| Advantages | Disadvantages |
|---|---|
| Easy understand and simple to use | High risk and uncertainty |
| Defined each of stage | Difficult measure the progress |
| Easily arrange the tasks | No suggest using for the long project |
| Phases are processed and completed at one time | Cannot accommodate changing requirements |
| Easy to manage and each phase have specified review process | If the problem is not solved completed will be fact many problems to next phase |

Table 2. 2: Major advantages and disadvantages of the waterfall model

2.4.2 Spiral Model

The second SLDC to be introduced is the Spiral Model. Spiral Model is the model combined with the sequential linear development model and iterative development process model. The idea of this model is combined with the iterative model with systematic, controlled aspect of the waterfall model. Waterfall model is a high emphasis on risk analysis, can incremental refinement through each iteration around the spiral.

This Model is widely used in the software industry in sync with the natural development process. The model is learning with maturity, involves minimum risk for the client. The model is most appropriate for long-term projects. Significant change is

expected during the development cycle, due to the clients are not sure the requirements and may involved need to evaluate to get clarity. Is suitable for budget constraint and risk evaluation for the project of medium high-risk.

2.4.2.1 Spiral Model Design

Spiral model has four phases. The model called Spirals because each project will repeatedly pass through these phases in iterations. The iterations of those of the process will be along to continues throughout the life of the software.

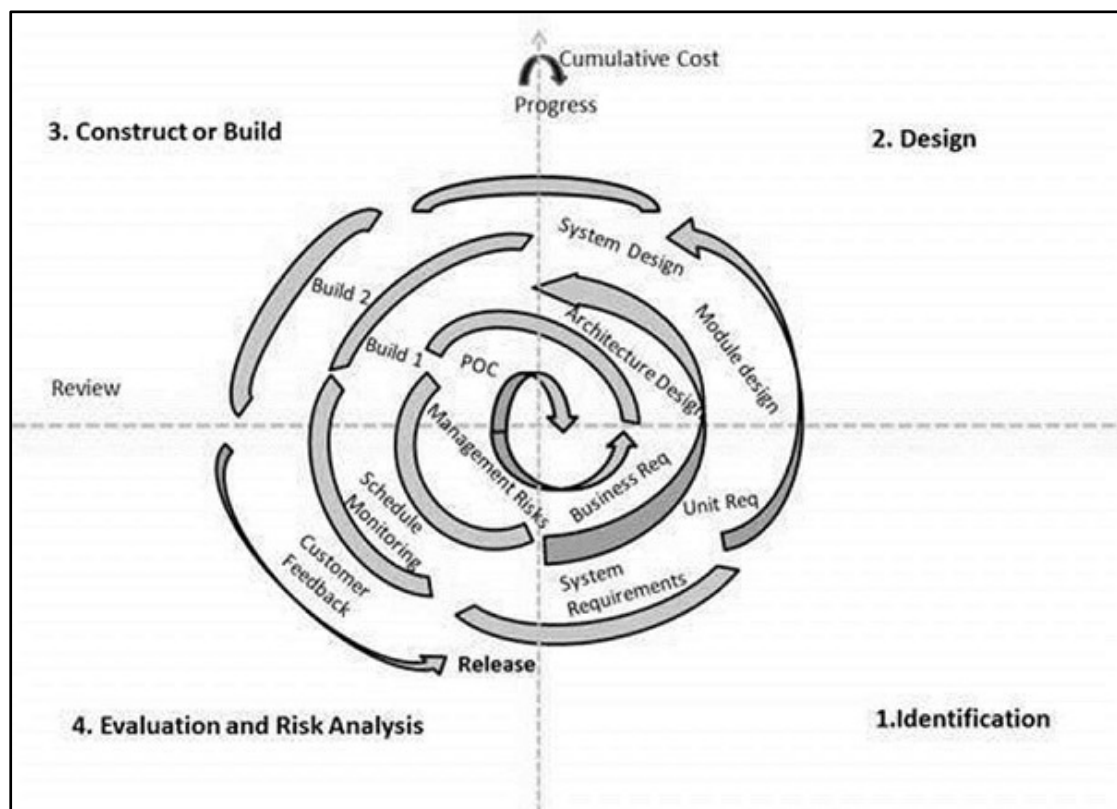


Figure 2. 9: Spiral Model (Spiral, n.d.)

The figure 2.9 is illustrated by the Spiral Model, list the activity of each phase.

2.4.2.1.1 Identification

Identification phase starts with the gathering of the requirement in baseline spiral. The subsequent spirals as identification of system requirements, unit requirements, product matures, subsystem requirements are done in this phase. Identification phase also includes continues to communicate between the customer and system analyst for understanding the system requirement. Example, discussing with customer and analyst the requirement should be included in the project. Requirements will be documented for next time discussing purpose.

2.4.2.1.2 Design

In the Design phase, beginning with conceptual design in baseline spiral and includes the logical design of modules, architectural design, physical product design and final design in subsequent spirals. Example, create the several types of design model and involve the logical design. All resources are from the Identification phase, the document.

2.4.2.1.3 Construct or Build

Construct phase are referring to the production of actual software project at each spiral. In baseline spiral, when a product just being design and developed of a POC (Proof of Concept). Is developed to get customer feedback on this phase. The Build is produced with a version number, is in the subsequent spirals with higher clarity on the requirement and details of designing a working model of software. All the builds will have sent to the customer for feedback. Example, based on the documented and the mock-up design, develop a system and send to the client for the feedback. The feedback will return to the developer, build and improve the previous system. The system will have a different version number.

2.4.2.1.4 Evaluation and Risk Analysis

In this phase, Risk Analysis contains estimating, identifying, monitoring technical feasibility and management the risks. Such as cost overrun and schedule slippage. After the testing of the build, in end of the first iteration, the customers are evaluating the software and provides feedback. Example, this phase will include evaluating how the system is work will or may facing some issues, the risk analysis will provide the report as the cost may overrun for the development. Each time will return the client feedback and improve the performance again.

2.4.2.2 Spiral Model Advantages and Disadvantages

The briefing of this model of pros is allowed the elements of product to be added, also force the early user to involve in the system development effort such as feedback from the customer. On the other hand, the brief of the cons is rigorous management complete; such product may have the risk of running spiral in the infinite loop.

| Advantages | Disadvantages |
|--|-------------------------|
| Allow widespread use of prototype | Process complex |
| Requirement be able to more accurately | Management more complex |

| | |
|--|--|
| Users are allowed to see the system early | Possible to run indefinitely loop |
| Can accommodate of changing requirement | Require excessive documentation for the intermediate stage of a large number. |
| Development be able to divide become smaller parts, risky parts can develop earlier for helping better risk management control | No established controls for moving cycle. If without the control, each cycle may be possible to generate more work to next cycle |

Table 2. 3: Major advantages and disadvantages of Spiral model

2.4.3 V-Model

Next, will introduce the SLDC is the V-Model. The reason why the model will be called as V because of the model when processing in sequential as V-shape. Beside that also known as Verification and Validation Model. This model is extended version of deriving from the Waterfall Model. Also, mean that each phase of the development cycle will direct associated testing phase. Before beginning, the next phase must be completed the previous phase.

V-Model is more or less similar to Waterfall model because both models are the sequential type. This model needs very clearly documented and no any ambiguous requirements. The project will be expensive if make a change and go back to the previous phase. Commonly will be used for the short-term project.

2.4.3.1 V-Model Design

The V-Model, two-phase will be corresponding planned in parallel. Testing phase and Development phase. To present V-shape, Verification phases and Validation phase both on a different side. In the end, Coding phase joins the two sides of V-model.

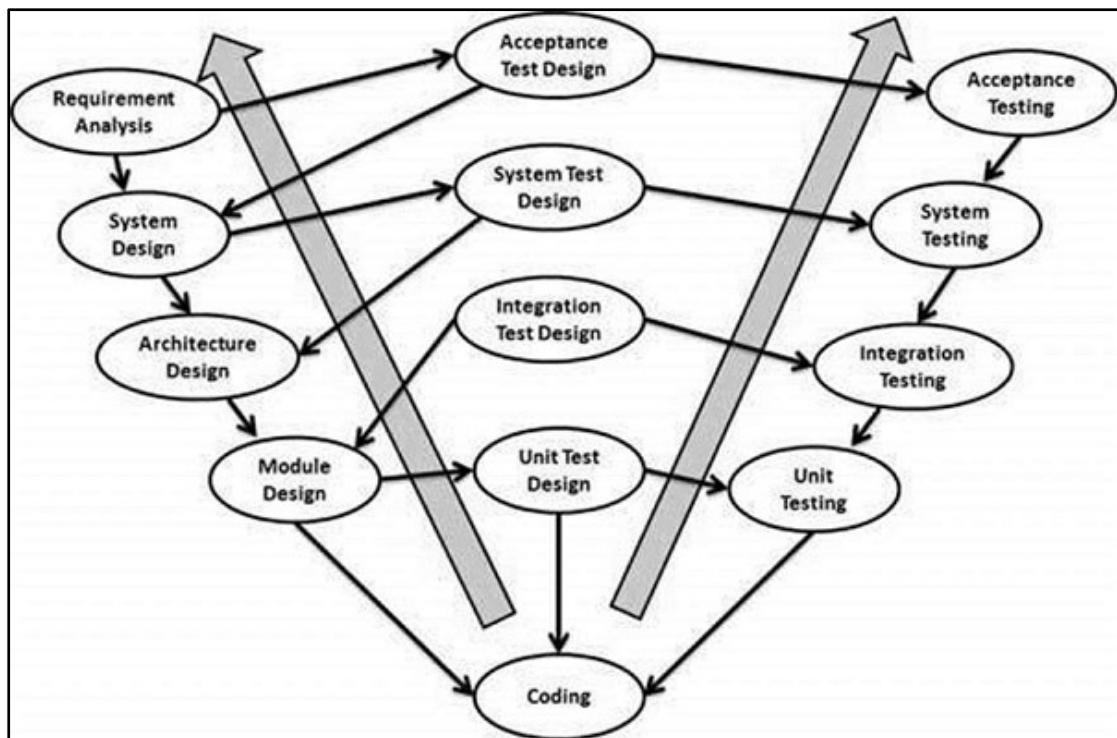


Figure 2. 10: V-Model (VModel, n.d.)

2.4.3.1.1 Verification Phase

2.4.3.1.1.1 Business Requirement Analysis

Business Requirement Analysis is the first phase of the development cycle. This phase requirements of the product understand from client's perspective. In this phase, is essential to communicate with the customer, to understand what his requirement and expectations. The business requirement can use an input for acceptance testing. Example, communicate with the customer requirements, understand and analyse the requirement and documented what customer wants.

2.4.3.1.1.2 System Design

In this phase, will understand the detail of hardware, software for the product under development. System test plan is developed on system design, and doing earlier will leave extra time for later actual test execution. Example, from the documentation, translate into specifying part of development requirement.

2.4.3.1.1.3 Architectural Design

This phase will design and understand the specification of the architectural. Not only one approach of technology is proposed and taken in the final decision. HLD (High-Level Design) as the System design split into modules with different functionality. This stage will be understanding and define the data transfer and communication between

internal model and external (other systems). This information can be used for integration tests, design and documented. Example, translate the document into specifying part of High-Level Design, this design can be used for later phase, the purpose is to understand what of the model and the functionality.

2.4.3.1.1.4 Module Design

LLD (Low-Level Design), specifying all system model of the detail of the internal design. This process is critical to ensure that design is compatible with other modules in system architecture and other external systems. Unit tests are designed at this stage for the internal module design. Is an essential part of the process to eliminate maximum error and faults. Example, Low-Level Model will be created, be ensure all the submodule can be compatible with each other for the integration.

2.4.3.1.2 Coding Phase

In this phase, will be using the best appropriate programming language was decided from the design phase. Coding performed are based on guidelines and standards to avoid unnecessary fault or error. This code will be reviewed and optimised for best performance before the final build. Example, use the suitable programming language to develop the system. Review and optimise the code for the best result.

2.4.3.1.3 Validation Phases

2.4.3.1.3.1 Unit Testing

This phase is testing all the code to reduce bugs in the early stage, though all of the defects cannot expose by unit testing. Example, review all the code and testing for the error, to eliminate the bugs in the early phase.

2.4.3.1.3.2 Integration Testing

Integration test is to performed to test the co-existence and communication of the internal modules within the system. Example, take the all submodule to integrate into one, testing after the integration to avoid late error or bug occurs.

2.4.3.1.3.3 System Testing

In the System Testing, is to check and test the entire system functionality and communication the system under development with external systems. Hardware and

software compatibility issues can be exposed during this testing process. Example, after the integration process, testing all the functionality is work well and test which hardware and software are not able to support.

2.4.3.1.3.4 Acceptance Testing

In this testing, can be exposed the compatibility issues with another available system in the user environment. The testing process also will discover the non-functional issues like reading and load performance of the defects in the user environment. Example, take the final system to test for the compatible on user environment and probably can find out the non-functional issues.

2.4.3.2 V-Model Advantages and Disadvantages

The briefing of this model is more accessible to managed and understand however this model is not flexible to make any change.

| Advantages | Disadvantages |
|---|--|
| Easy understand and simple to use | High risk and uncertainty |
| Defined each of stage | No suggest using for the long project |
| The testing process will involve in requirement phase | Not useful model for complex projects |
| Phases are processed and completed at one time | Cannot accommodate changing requirements |
| Easy to manage and each phase have specified review process | Challenging to make change and back while in the testing stage |

Table 2. 4: Major advantages and disadvantages of V-Model

2.5 Development tools background

The Development tools are divided into two type which is classified as software and hardware. The constraints are the list of software and hardware that used or involved in the video game development. It is divided into two categories which are software and hardware. In software will be categories into three type, such as development, designing, version control (backup).

2.5.1 Development

In Development category, this category of the software will work more toward on the developing.

2.5.1.1 Unreal Engine

Unreal Engine 4 is a complete suite of tools to designed meet aspiring artistic visions while being elastic sufficient to make sure success for teams of all sizes. Unreal provide convincing performance as an established, industry-leading engine. Breaking the barriers with tool and workflows put in control. Unreal's provided pre-built, customizable plugins, modular systems and source control integration enable the developer to meet every single project's unique needs. Unreal has pushed the limit of visual experience, by choosing of the highest calibre of professionals. Unreal provided a community to share a passion for creation, question, commitment and to achieving excellence. The members are eligible to join these communities are aspiring developer or a seasoned veteran. Epic, the company, uses the engine ourselves to award-winning craft content resulting in powerful tools, and pipelines facilitate developer to accomplish goals. Unreal has become trusted, reliable engine in the world from over the two decades. The Unreal Engine is complete product suite to production-ready out of the box without for additional plugins or purchase. Three Main Feature of Unreal Engine

2.5.1.1.1 Blueprints: Create without Coding

Blueprint, is friendly design with visual scripting, develop can quickly prototype the code and send interactive content without touching a line of code. The blueprint can be used to modify user interfaces, build object behaviours and interactions, adjust input controls and more. The blueprint can be allowed to visualise gameplay flow and inspecting properties while testing work using the powerful built-in debugger.

2.5.1.1.2 Full C++ Source Code Included

Unreal Engine, the primary programming language to be used with this tools are C++. With complete C++ source code access, the developer can study, customise and debug the entire Unreal Engine, and ship the project without obstruction.

2.5.1.1.3 Photoreal Rendering in Real Time

Unreal Engine has achieved the Hollywood-quality visuals out of the box. Unreal Engine's physically-based rendering, screen space reflections, advanced dynamic shadow options and lighting channels provide the flexibility and efficiency to create excellent content.

2.5.1.2 Visual Studio

Microsoft Visual Studio community 2017 is an integrated development environment (IDE), developed by Microsoft. This program is used to develop computer software or program for Microsoft Windows, mobile apps, web apps, websites and web services. Visual Studio uses Microsoft software development platforms like Windows Store, Windows Forms, Windows API, Windows Presentation Foundation, and Microsoft Silverlight. The programme can manage code and produce native codes.

Visual Studio included IntelliSense (a code editor support to complete the code component) to do the code refactoring. The integrated debugger works as a machine-level debugger and a source-level debugger. The built-in tools include web designer, class designer, code profiler, forms designer for building GUI applications, and database schema designer. Visual Studio is accepting plug-ins to enhance functionality at every level—including adding new toolsets such as editors and visual designers for domain-specific languages or toolsets for further aspects of software development lifecycle (like Team Foundation Server client: Team Explorer) and adding support for source control systems (like Subversion). The Main Feature of Visual Studio

2.5.1.2.1 Develop: Navigate, write, and fix your code fast.

Visual Studio supports the developer to write code precisely and efficiently without losing the current file context. The program can quickly zoom into details such as related functions, call structure, check-ins, and test status. Visual Studio also leverages our functionality to identify, refactor, and fix code issues.

2.5.1.2.2 Debug: Debug, profile, and diagnose with ease.

The debugger of Visual Studio is quick to find and fix the error bugs across languages, locally or remotely, and with historical data of code. Leverage profiling tools to discover and diagnose performance issues without leaving debugging workflow.

2.5.1.2.3 Test: Write high-quality code with comprehensive testing tools.

The test tools are helping the developer to deliver high-quality software. Use these tools to plan, execute, and monitor the entire testing effort. Stay on the top of test plan with quality metrics, indicators, and comprehensive test status reporting.

2.5.1.2.4 Collaborate: Use version control, be agile, collaborate efficiently.

Visual Studio is provided with the Team Service to manage the code with alongside bugs and to work with the items for the whole project. This program also can be managing source code in Git repos hosted by any provider, including GitHub.

2.5.1.2.5 Extend: Choose from thousands of extensions to customise your IDE.

Visual Studio is functionality because the program is taking advantage of the tools, controls, and templates available from Microsoft, community and the partner of Microsoft. The user can customise to liking by building own extensions.

2.5.2 Designing

In the Designing category, the software's will more focus on building the Mesh Object, create or modify the texture of the image and the sound correction.

2.5.2.1 Blender

Blender 2.79 is an open source 3D creation suite and free to use. This software supports the entirety of 3D pipelines such as modelling, rigging, animation, simulation, rendering, compositing, motion tracking, video editing and game creation. The advanced users are employed in Blender's API for scripting Python language to customise the application and create specialised tools. The future release version of Blender is often to include these tools. Individuals and small studios benefit from the unified pipeline and active development by using Blender. Examples from many Blender-based projects are available in the showcase. The Six of Main Feature of Blender,

2.5.2.1.1 Photorealistic Rendering

Blender provided a dominant feature with new unbiased rendering engine called Cycles; this offers beautiful ultra-realistic rendering. Such as Real-time viewport preview.

2.5.2.1.2 Fast Modeling

Blender provided several tools to a comprehensive array of modelling tools make creating, transforming and editing models a breeze. The tools such as Keyboard shortcuts for a fast workflow.

2.5.2.1.3 Realistic Materials

The Blender with the new rendering engine is possibilities for materials are endless. Such as Complete Node Support for full customisation.

2.5.2.1.4 Fast Rigging

Blender offers an impressive set of rigging tools including Envelope, skeleton and automatic skinning. This can be done by quick of the rigging process.

2.5.2.1.5 Fast UV Unwrapping

Blender is provided with a way to unwrap the mesh quickly, and use image textures or paint own directly onto the model. Blender allows this feature like Fast Cube, Cylinder, Sphere and Camera projections.

2.5.2.1.6 Amazing Simulations

Blender delivers a great looking result of the simulations, such as rain, smoke, fire, crumbling building, fluid, cloth or full of destruction.

2.5.2.2 Adobe Fuse

Adobe Fuse CC (Beta) is desktop software to allows creatives of all levels to build highly custom, high-quality 3D character models in quick. Adobe Fuse characters are highly robust and can use in a variety of projects such as prototyping design layouts, graphic design, staging a video shoot, or creating interactive video games.

Adobe Fuse can be used by a user who is without any prior 3D experience. The user can quickly build characters using build it a library of high-quality 3D content. From faces and bodies to clothing and textures with an option to customise texture, colour and shape. Designer or User can easily save the project characters to Creative Cloud Libraries and bring them into a 3D workspace in Adobe Photoshop CC to make realistic photo composites with 3D models.

2.5.2.3 MakeHuman

MAKEHUMAN 1.1.1 is free to use and open source software to create a realistic 3d mesh of humans for using in illustrations animations games of Zbrush/Mudbox sculpting. The content created by using MakeHuman is licensed under the CC0 license, giving artists the unprecedented freedom to creations in any way. As many of parameters or built-in feature are free to download and use for free. The code,

knowledge, the algorithms and technology developed by MakeHuman team are freely shared with the world. Thousands of morphings for efficient parametric modelling Easy and intuitive parameters, including gender, Age, weight, height, body proportions, face shapes, eyes, nose, mouth, chin, ears, neck and more. MakeHuman and Adobe Fuse are similar software; both are used to generate temporary character mesh and import into Blender to do modification and improvement.

2.5.2.4 Adobe Photoshop

Adobe Photoshop CC is leading of photo editing and manipulation software on the market now. This software is used from the range from the full-featured editing of large batches of images to creating intricate digital paintings and drawings mimic those done by hand. The Main Feature of Adobe Photoshop,

2.5.2.4.1 Easy design tools

Adobe Photoshop is providing the tools to easy-to-use that help user creates beautiful images for design projects like flyers, banners, websites, and more.

2.5.2.4.2 Professional photography tools

The tools are as necessary to advance to use, such as remove objects, combine multiple images and retouch photos by using the complete set of professional photography tools.

2.5.2.4.3 Easy access to Adobe Stock assets

This software is quick to crop the image with Content-Aware technology to preserve the looking of the image and create precise portrait adjustments with Face-Aware Liquify.

2.5.2.5 Adobe Illustrator

Adobe Illustrator CC is software to use by both graphic designers and artists, for creating vector images. These images will be used for such as company logos, both in print and digital form are not be readily jaggiest. This software is typically used to create logos, illustrations, graphs, charts, diagrams, cartoons of real photographs, and more. The Main Feature of Adobe Illustrator,

2.5.2.5.1 Pixel-perfect artwork

Create an image with pixel-perfect artwork for screen designs by drawing shapes and paths that seamlessly align with the pixel grid.

2.5.2.5.2 Artboard enhancements

Adobe Illustrator can be able to select an entire artboard or choose individual assets from one or other artboards, and export to multiple sizes, resolutions, and formats in a single click.

2.5.2.5.3 Presets and Adobe Stock templates

This software is pre-build a template to make the designer design faster with presets and templates for brochures, business cards, and more.

2.5.2.6 AwesomeBump

AwesomeBump 4.0 is an open source program and free to use as designed to generate normal, specular, height, or ambient occlusion, metallic and roughness textures from a single image. Due to the image processing is done in 99% on GPU, the program is run as very fast, and all parameters be able to change in real time with real-time preview display. AwesomeBump was made as an alternative of Insane Bump (A GIMP build-in). Since version 3.0 AwesomeBump is supporting PBR(Physically-based rendering) shading model which makes the software better and version 4.0 Grunge maps are available.

AwesomeBump is written in Qt, thus don't need to install any additional libraries. Download and install Qt SDK, download the project from the repository, build and run. The program will work (or should) on any platform supported by Qt. The Awesome Bump can do such as

- 2.5.2.6.1 convert from normal map to height map.
- 2.5.2.6.2 Convert from height map (bump map) to normal map.
- 2.5.2.6.3 Extract the bump from the arbitrary image.
- 2.5.2.6.4 Calculate ambient occlusion and specularity of the image.
- 2.5.2.6.5 Perspective transformation of the image.
- 2.5.2.6.6 Creating seamless texture (simple linear filter, random mode, or mirror filter).
- 2.5.2.6.7 Generate roughness and metallic textures (different types of surface analysis are available).
- 2.5.2.6.8 Real-Time tessellation is available.
- 2.5.2.6.9 Saving images to following formats: PNG, JPG, BMP, TGA.
- 2.5.2.6.10 Edit one texture which contains different materials.
- 2.5.2.6.11 Add some grunge to your map with grunge texture.
- 2.5.2.6.12 Mix two bump maps with normal map mixer and more.

2.5.2.7 CrazyBump

Crazybump is an easy and quick tool for creating normal maps from the texture. The default settings, however, are hardly ideal for any texture and are meant as a starting point for normal map creation.

AwesomeBump and CrazyBump both are similar software; it allows to import and export to another texture map, such as diffuse, height map, bump map, normal map and more.

2.5.2.8 Audacity

Audacity is free to use software, developed by a group of volunteers and distributed under GNU General Public License (GPL). Audacity is easy-to-use, multi-track audio editor and recorder for Windows, Mac OS X, GNU/Linux and other operating systems. The interface is translated into many languages. Audacity can be used for

- 2.5.2.8.1 Record live audio.
- 2.5.2.8.2 Record computer playback on more platform.
- 2.5.2.8.3 Convert tapes and records into digital recordings or CDs.
- 2.5.2.8.4 Edit WAV, AIFF, FLAC, MP2, MP3 or Ogg Vorbis sound files.
- 2.5.2.8.5 AC3, M4A/M4R (AAC), WMA and other formats supported using optional libraries.
- 2.5.2.8.6 Cut, copy, splice or mix sounds together.
- 2.5.2.8.7 Numerous effects including changing the speed or pitch of a recording.
- 2.5.2.8.8 Write your plug-in effects with Nyquist and more.

2.5.3 Version Control (Backup)

This category of the software is mainly to do as a backup, or another feature is version control. Version Control it allows the user to commit (backup) every time the file with the comment. It also allows the user recall back any stage of use wants to.

2.5.3.1 Git

Git is version control system for tracking the changes in computer files and synchronising work on these files among multiple people. The primarily used for source code management in software development, however, can be able to use in any file, keep track of change. As distributed revision control system is aimed at speed, the integrity of data, and support for distributed, non-linear workflows. Git was created by Linus Torvalds in 2005 for development of Linux kernel, with other kernel developers contributing to initial development. The current maintainer since 2005 is Junio Hamano. As most other distributed version control systems, and different most client-server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server. Git is free software distributed under terms of GNU General Public License version 2.

2.5.3.2 GitHub

GitHub is web-based Git or version control repository and Internet hosting service. This is mostly used for code, offers all of distributed version control and source code

management (SCM) functionality of Git and adding own features. To provide access control and several collaboration features such as feature requests, bug tracking, task management, and wikis for each project. GitHub offers several plans for private and public (free) repositories on the same account which commonly used to host open-source software projects. In April 2017, GitHub reports the application are having almost 20 million users and 57 million repositories, making the most important host of source code in the world. GitHub has a mascot called Octocat, a cat with five tentacles and a human-like face.

2.5.3.3 GitHub Desktop

GitHub Desktop is client software on git and GitHub. Easy to use. The Main Feature of this Client software is

2.5.3.3.1 Experience Electron

GitHub Desktop are providing the simplified experience with more space of code for the different operating system.

2.5.3.3.2 Streamline your workflow

GitHub Desktop can create branches, commit changes without touching the command line and collaborate with other developers.

2.5.3.3.3 Make it your own

Every pixel is entirely free and open source. Build the features are need to use and be part of future GitHub Desktop releases.

2.5.3.4 Sourcetree

An alternative software of Github Desktop is free Git client for Windows and Mac. Sourcetree simplifies how the user to interact with Git repositories so can be a focus on coding. Visualize and manage the repositories through Sourcetree's simple Git GUI. The software is similar to GitHub Desktop/Client. The Main Features of this software,

2.5.3.4.1 The new Git status

The user is Never missing a thing. Stay on top of work and up to date with code at a glance.

2.5.3.4.2 Visualize your progress

The Detailed branching diagrams make the work easy to keep up with team's progress.

2.5.3.4.3 Git for Windows & Mac

The SourceTree is free to use for different Operating System and Wield the power of Git and Mercurial on two most popular operating systems.

2.5.3.4.4 Get Git right

The user can quickly learn Git through comprehensive tutorials covering merging, branching and more.

2.6 Conclusion

In this chapter is more about the background of the existing game, software and hardware to use, and several methodologies. This chapter can send the message and content to the audience why the “You Are Not Alone” will be created and which game is inspired by me. What is the reason and what is the improvement from the existing game after comparison? What is SDLC means and discussing in detail about several types of model, in each process?

In conclusion, even simple types of game can be designed to address specific learning outcomes such as recall of factual content or as the basis for active involvement and discussion (Blake & Goodman, 1999) (JV, et al., 1996). Computer games: (Prensky, 2001) states that computer games can be characterised by six vital structural elements which, when combined, vigorously engage the player. These elements are rules, conflict/competition/challenge/opposition, goals and objectives, interaction, outcomes and feedback, and representation or story.

CHAPTER 3

DEVELOPMENT METHODOLOGY

3.1 Introduction

Software or System Development Methodology also as Software Development Life Cycle (SDLC). SDLC is called as Software Development Process. An SDLC (Software Development Life Cycle) follows the crucial phases that are important for developers and the team, like planning, analysis, design, and implementation. According to Ruparelia (2010), each Software Development Life Cycle can classify into several types, such as linear, iterative, or combination of both (Linear with Iterative). The type of Linear model follows the sequent. It means that every single stage automatically leads to the next stage and is not repeated or rolled-back. Next, the type of Iterative models is different from the Linear type, it is a continuous process, and each stage can be revisited multiple times during the life of the project. Last, the combined model is an iterative mode with the finish line and is not necessary to revisit any of the stages.

In this chapter, we will discuss more which methodology is suitable for the project. The reason why the methodology is suitable and how each of the processes works nicely for the project.

3.2 Methodology chosen

3.2.1 Comparison Methodologies (Waterfall, Spiral, V-Mode)

| | Waterfall | Spiral | V-Mode |
|--|------------|-----------|------------|
| Flow Type | Linear | Iterative | Linear |
| Well Defined Requirement | Yes | No | Yes |
| Availability of Reusable Components | No | Yes | No |
| The complexity of the System | Simple | Complex | Simple |
| Duration of project | Short-term | Long-term | Short-term |
| Domain Knowledge of Team member | Adequate | | Adequate |
| The expertise of Users in Problem Domain | Very Less | Very Less | Very Less |
| User involvement in all SDLC Phase | No | No | No |

Table 3. 1: Comparison of Waterfall, Spiral, V-Model

Based on table 3.1, be able to describe the significant part of these methodologies. The detail for this three methodology will be discussed in the previous chapter (Chapter 2, Development Methodology Background).

The first methodology to briefing discuss is the Waterfall Model. This model is a traditional and simplified version of SDLC for the software engineering. Waterfall model is the classic approach, and the development method of flow is rigid and linear. Waterfall model is to explicitly set the goal in each phase and completed before the move to the next phase. This model will have divided into several phases, however, connect as a sequent line. Planning, time schedules, target dates, budgets and implementation of the system at once is the importance. The control is tight for maintained over the project through the use of extensive written documentation.

The second methodology to discussing is the Spiral Model. This model mainly to focus on early identification and reduce the risks of the project. A project of the spiral model will keep the risk, makes a plan to solve the risk and decides when to take next

phase of the project. This methodology does not increase the project processing speed, however, be able to reduce the risk level of the project. In another word, Spiral Model use to breaking down the project into the smaller segment and provide ease-of-change during the development process, as well as providing an opportunity for the project to eliminated the risks.

Lastly, the methodology is the V-Model. The V-Model or V-Shape is the modified version of Waterfall Model. This is the extension of Waterfall Model however not designed in Linear flow, instead of the stage turn back upward after the coding phase is done. The project to use this methodology has to be check from each phase and approved to move forward. This model is developer and tester work parallel, is stable and depend on verification from the previous phase before going on forward. V-Model, provide the relationship between every stage (Development and Testing).

From the table 3.1, Waterfall and V-Model is the framework of Linear. However, Spiral is Iterative. Because the V-Model is the extension of Waterfall and Waterfall is the linear type of flow. Waterfall Model is work in the sequential order, but the Spiral Model is processing on looping. Waterfall Model and V-Model are requiring the clearly and fix documentation, however Spiral Model not necessary, because Spiral Model is flexible to change the requirement during development processing. Waterfall Model and V-Model are inappropriate to reuse the component, but Spiral Model is free to use. From the short brief of above, be able to determine the complexity of the project from simple to complex, and from short-term to long-term.

In short, the methodology to be used for this project is Waterfall. Because the requirement of this project is explicit before development begins. To confirm the project not spend much time, each phase must be completed before the move to next phase. Compare to Spiral Mode, have the possibility of an indefinite loop, instead of a linear model, easy to implement. Move over, the project is not the team-based, is suitable for selected waterfall model, not V-Model. Because V-Model requires the tester to testing during the processing, is involving much of the user. However, this process may take much time compare with waterfall model. The progress of the system development is measurable and is suitable for the short-term project. In this project may possible have any change during the development process. However, this process can be done by after the maintenance phase. The entire system can be divided into several

smaller part, and each of the parts can be created and use for every single of the waterfall model. Meant that, each smaller part of systems is starting by first phase (Requirement Analysis) of the Waterfall Model. More comfortable to carry out and review the progress.

3.2.2 Waterfall Model

The example of the sequential model is the Waterfall Model. Waterfall Model is the classic approach, and the activity of the software development is to divide different phases and each phase consist of series tasks. The reason why this model named as waterfall because of the development process like a waterfall. From higher level to lower level, the first phase must be completed before the move to next phase.

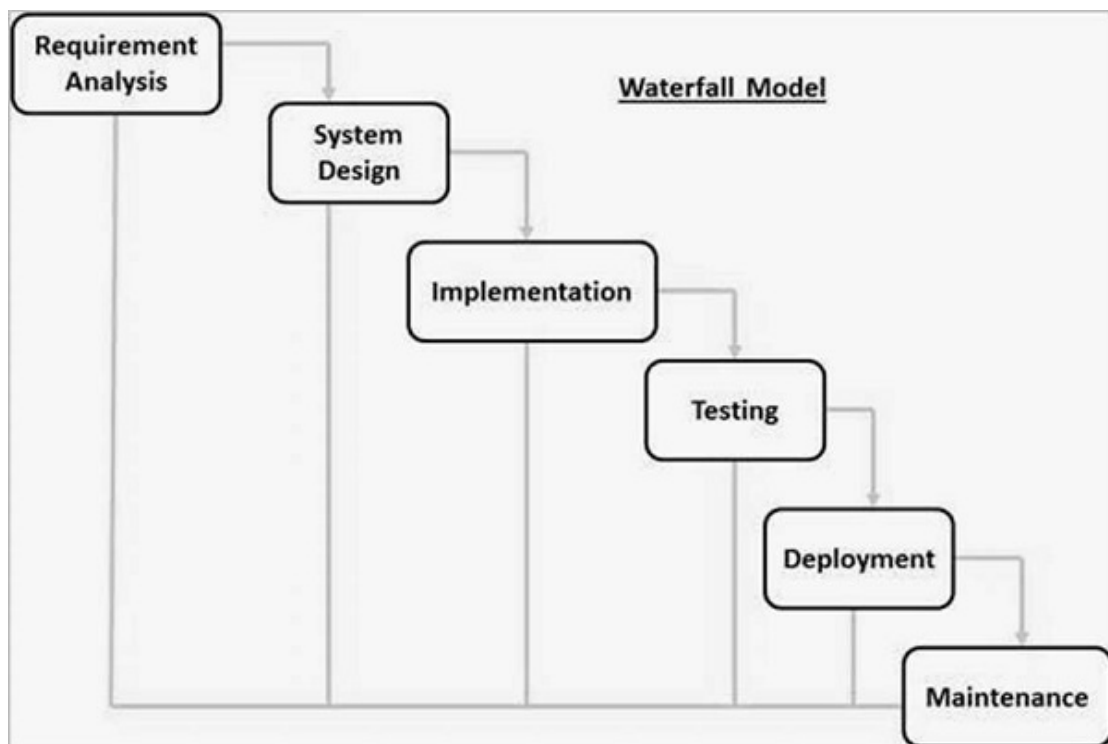


Figure 3. 1: Waterfall Model (Waterfall, n.d.)

3.2.2.1 Requirement Analysis

The first phase of the Waterfall model is Requirement Analysis. The Requirement process, during the initial phase, every potential requirement is analysed and documented in the specification document, to serve for all future development. The outcome of this process is defining what application should do, however not how to do. For the Analysis, is to identify to generate models and business logic that will use in

the application. In this phase, the activity can be captured all the requirement are needed by doing the brainstorming and walkthrough to understand the requirement. To check the feasibility of the requirement is testable or not. In the end, will deliver the RUD (Requirement Understanding Document).

In this phase, “You Are Not Alone” project is gathering the information from the existing game information, the comment of the user from the existing gameplay, and the suggestion for improvement from the existing game. From the Chapter 2, section of Research/Findings on the Existing System had been mention, why “You Are Not Alone” project is planning to create and which existing game is inspired.

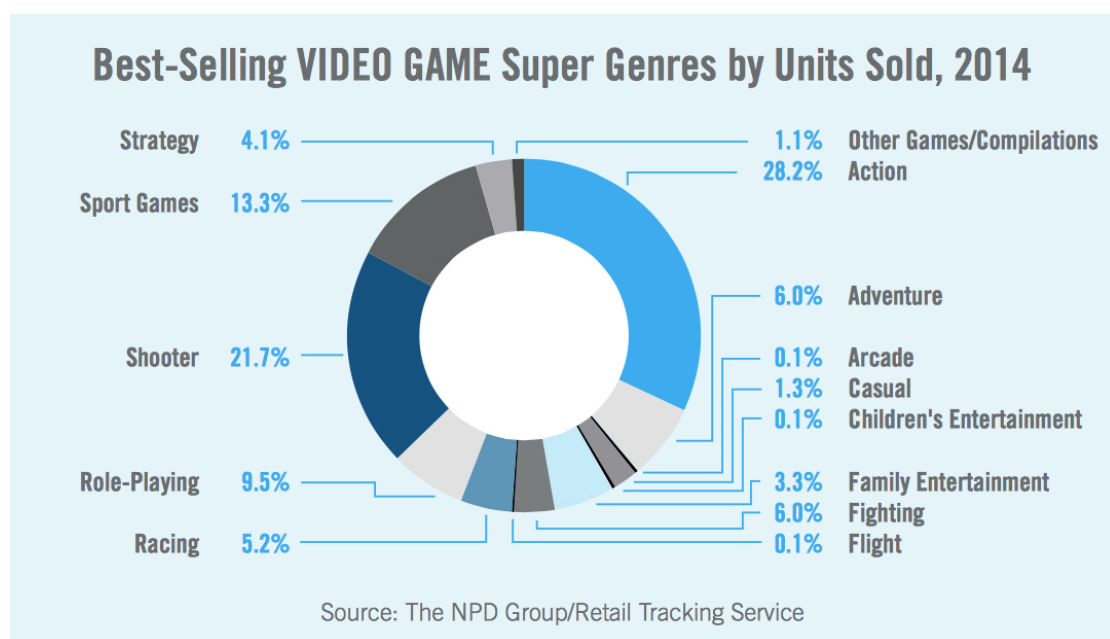


Figure 3. 2: Best-Selling Video Game Genres in 2014 (Best Selling Video Game, 2014)

Based on figure 3.2, this image is presented what is the best genres of the game are sold in 2014. The project “You Are Not Alone” is involved several types of genres, but in short, the game is choosing to use the viewport as FPS (First-Person Shooter), one of the reason is the game genre of Shooter contain a massive amount of the percentage in the favour game in the world. The player is like to play the shooting game one of the reason because FPS view is quite realistic. The project is involving another genre such as Strategy, Adventure, Action. Some user is like to play the strategy game, mean that try to solve the problem to pass through a next stage. In Adventure, the gamer is like to

explore the world, in action the game more toward the movement of animation. To create a game with several types of genre, Horror genre is contained all of these.

In this phase, the project is collect the information from the existing game, such as which building are better to use in horror game. The best building user is like to play in the game while the game is horror type is hospital, school, church and more. More of the horror game is a limited resource such as battery requires using a flashlight. Jump scare in arbitrary timing is better than just on the particular place will activate the jump scare. The game more to hide and seek are significant to a fighting game. In the end, different ending scene is attractive the player to play more than one time and understand the story more precisely. The information also collected from the questionnaire, in Chapter 4 will discuss more detail.

3.2.2.2 System Design

In this phase, the requirement specifications from the first phase (Requirement Analysis) are studied, and the system design is to prepared. Is now to implement with the help of these requirement specifications. System Design phase needs to decide the information of architecture and the technology applied in the project, like what is the programming languages to use, which class libraries, and how the program sequences. The outcome of the design is commonly documented in the diagram to describe the software of the general behaviour. Usually, the activity performed in this phase is to create the design as each of the requirement, based on the requirements necessary to capture the hardware and software use. In the end, document the design. The deliverable result with have two, HLD (High-Level Design Document) and LLD (Low- Level Design Document).

System Design phase is to help the Requirement Analysis phase to decide which of the requirement are need to use and which are not. In the project “You Are Not Alone”, the software and hardware have to use have been documented in detail in Chapter 2, Section Development Tools Background. On the section, be able to understand the details of the software. The game engine is used to create the project is Unreal Engine, this engine is the best to create Video Game in ‘AAA’ standard. The engine is requiring to use the programming language is C++. To require in this language are fully compile into the game engine is to use Visual Studio IDE.

Most Popular 3D Modeling Software for 3D Printing

| Rank | Software | General | | 3D Printing Community | | | | Total Score |
|------|---------------------|---------|---------|-----------------------|---------|--------|--------|-------------|
| | | Social | Website | Forum | YouTube | Models | Google | |
| 1 | Blender | 67 | 89 | 97 | 100 | 57 | 92 | 83,7 |
| 2 | SketchUp | 93 | 82 | 100 | 96 | 52 | 76 | 83,2 |
| 3 | SolidWorks | 99 | 77 | 92 | 97 | 46 | 76 | 81,2 |
| 4 | Fusion 360 | 96 | 85 | 53 | 95 | 15 | 99 | 73,8 |
| 5 | Inventor | 95 | 81 | 61 | 70 | 11 | 100 | 69,7 |
| 6 | Maya | 93 | 85 | 15 | 94 | 3 | 100 | 65,0 |
| 7 | AutoCAD | 100 | 84 | 27 | 93 | 7 | 73 | 64,0 |
| 8 | 3DS Max | 97 | 84 | 47 | 91 | 4 | 60 | 63,8 |
| 9 | ZBrush | 87 | 67 | 39 | 90 | 5 | 66 | 59,0 |
| 10 | TinkerCAD | 91 | 74 | 48 | 21 | 100 | 19 | 58,8 |
| 11 | Cinema4D | 89 | 73 | 44 | 9 | 7 | 64 | 47,7 |
| 12 | Rhinoceros | 32 | 72 | 52 | 39 | 12 | 66 | 45,5 |
| 13 | OpenSCAD | 2 | 66 | 51 | 3 | 95 | 54 | 45,2 |
| 14 | MODO | 85 | 48 | 10 | 10 | 1 | 37 | 31,8 |
| 15 | Meshmixer | 4 | 58 | 41 | 13 | 23 | 5 | 24,0 |
| 16 | Sculptris | 44 | 65 | 10 | 3 | 6 | 4 | 22,0 |
| 17 | PTC Creo Parametric | 36 | 65 | 6 | 7 | 3 | 14 | 21,8 |
| 18 | FreeCAD | 10 | 61 | 13 | 11 | 29 | 2 | 21,0 |
| 19 | Solid Edge | 47 | 45 | 4 | 2 | 1 | 26 | 20,8 |
| 20 | OnShape | 36 | 59 | 3 | 3 | 12 | 3 | 19,3 |

i.materialise

Figure 3. 3: Most Popular 3D Modeling Software 2017 (Most Popular 3D Modeling & Design Software for 3D Printing, 2017)

Based on figure 3.3, the most modelling software now is suggesting to use Blender. Blender is open source and free to use. The project is used to create several crucial 3D object mesh and Animation, like Building, Character, and other furniture equipment. In the game of the HUD (Head-Up Display), this is usually creating in the Unreal Editor software. The other reason why to use Unreal Engine is that easy-to-use of the software to create the HUD and implement the scene in the game. Unreal Editor HUD is allowing to create a dynamic display. Unreal Engine is also used to create the materials. In the

backup software will be recommended to use Version Control rather than just a simple backup application. Version Control is allowing to commit the file and backup each time; the recovery process can roll back each commit that was created. This is why it quickly reduces the time to find and solve the error that occurs.

The game of the design, such as the game is in a dark place. All are collected the information from the first phase, and this information is getting from the existing game of customers.

The game of the storyboard and interface design will display in Chapter 5.

3.2.2.3 Implementation

During this phase, workflows and structure are implemented into system framework condition and objectives. The software design document has become to use the program that directly related to OS (Operating System), and programming language, and the infrastructure. The outcome of this phase is the application, often as a beta version. With the input from the System Design, the first system is developed in the small program is called as Units. The activity to perform in this phase are create the programmes or code from each of design. All of the units will be integrated into next phase and test. This phase is delivered several of the units (had been tested) and prepared to be used for next phase.

“You Are Not Alone”, during implementation. The program is developed by several different projects. Each of the project or the level will be implemented just part of the feature. The way to do that is to avoid the system are crash easily and waste much of time to finding and debugging. In the project, implementation can be use or modify the design from the previous phase (System Design). Implement the 3D object in the game to test the measuring unit are compatible. Use the 3D mesh object to connected the code. Such as the movement of the Character, the effect of the door opened and more. This phase will be joining the particle effect and the character movement to make it more realistic effect in the game, like the dust out after the character feet on the floor.

In Unreal Engine, coding process can be used as C++ and the Blueprint feature. Blueprint feature is a way without key any programming code however to create the game. However, this is requiring the user have the knowledge or skill in programming, due to the reason why is because if the developer is ZERO standard knowledge of programming will facing difficulty. In short, the blueprint is the first way to testing the

function are working or not, the feature is allowing the developer to watch the code flow process and run the game at the same time. If the blueprint code is worked, the function will be reconstructed into the C++ class. The reason why C++ class is better than Blueprint because C++ has excellent optimistic rather than just all the code in the blueprint. However, some of the features are not recommend did in C++, such as Open the door. Because C++ will set the fixed angle to open the door, like Yaw rotation from 0 to 90. However, in Blueprint, is allow the door are dynamic to open from 0 to 90, every second is starting to plus one angle. In short, during this phase are create several features, test it and before integrate into a single program.

3.2.2.4 Testing

The testing phase is followed by testing of each software modules, components, and the whole system. Integration in this phase, if conflicts and errors occur, is required to repair immediately. Such could lead increase the time and budget to spend since the possible error can have attributed different phases and not always caused in the previous phase. To avoid the overrun time and overspent budget. The activity to perform in this phase is to integrate the unit and tested to ensure it works as expected. Functional and Non-Functional activities all have been tested and make sure the system meets the requirement. In case of an anomaly, report. Based on the report, track the progress and testing through tools like traceability metrics, ALM (Application Lifecycle Management). The outcome of this phase delivers test case, report (Test and Defect). Moreover, the result is the integrated system with thoroughly tested, prepared to deploy and launch in the market.

The process implements in the project, “You Are Not Alone”. Integrated all the feature, function, design into an entire system by using all the units from the previous phase (Implementation). The feature included character movement, inventory system, HUD (Head-Up Display), a 3D object such as House. Based on the Implementation phase, each unit had been tested before integrated together. The Post integration the whole system is tested for any failures and faults. To integrated these units, and another object. Mainly was used Unreal Editor to do that. Before integrated is the best way to backup and commit each of the files, if any problem occurs is allow the developer to rolled-back any commit (any version of the file had been a backup) without undoing the whole process or destroy the entire system or each unit. Recommended to back up the file by using GitHub.

3.2.2.5 Deployment

Once the entire system was tested (functional and non-functional), the product is to deploy into customer environment market or released into the market. Updates and maintenance may be necessary before the product enters a store or delivered to the customer. In short, the game is implemented after acceptance by customers. The activities in deployment phase are performed, ensure the environment is up and make sure no set defects open. Test exit criteria are met. Launch application in the respective environment. Moreover, to perform sanity check in the environment after the application is deployed to make sure game does not break. The result of this phase is to provide User Manual, Environment Definition or Specification and the Completed System.

The project is planning to deploy in the Global Digital Gaming Marketplace, such as STEAM. Steam is not just only selling the game; the platform can sell system (application), movie. The process to published the game on STEAM, first need to export and published the whole game by using Unreal Engine. To create the game become installer (setup), such as using Inno Setup. After that, Register the STEAM, is require Digital Paperwork, apps fees and the process. In the end, waiting for a STEAM pass, the game is sold to STEAM.

3.2.2.6 Maintenance

There may raise some issues which appearance client environment. To fix the issues, up-to-date version or patches are released. The updated version is enhancing game become better version are released. Maintenance is done deliver these change in the customer environment. In this phase, ensure the application is running and up in the respective environment. The issues face needs to fix in this phase. The updated code is deployed in the environment. Applications are keeping to update for enhanced incorporate more feature, update environment with latest features. The result is deliverable a new User Manual, new updated version of the system.

“You Are Not Alone” project is to maintained frequently after launch in the Digital Game Market. Based on the user or customer comment and suggestion, to fix the update the new patch in the game. That may be possible such as better materials. If the customer requires to more of this game, DLC (additional content). This phase is the

last phase of the Waterfall Model if the Additional Content is requiring to develop. The SDLC, the waterfall model, is requiring to create once again in the process.

3.3 Conclusion

Software development is a process of developing software through succeeding phases in an arranged way. This process includes not just writing of code, but the preparation of objective and requirement, the design of coded, and approval what was developed has met objectives.

Studied to understand different development models and their comparison. Masses of development models exist. This chapter explained three different models out of those. Waterfall Mode is the first, which provides a base for other development models. Next, enhanced models are explained. Spiral model, V-shaped model. The comparison which can help to decide which specific model is a demand to use in this project. In this chapter will be briefing descript each phase of the process related to the project. How the SDLC will connect to this game development. To understand more about Waterfall Model, classic SDLC model. With linear sequential method has set of objectives for each development phase. This model simplifies task scheduling because of no iterative or overlapping steps.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Introduction

System Analysis and Design is an active field which analysts repetitively acquire different approach and a new approach to building system more efficiently and effectively. The System Analysis and design of primary objective is improving the system.

System Analysis is to collect and interpret facts, identify problem and decomposition of the system into components. Besides that, is conducted to study system or the parts to identify the objective. Additionally, is a problem-solving technique to improves system and confirms all the components of the system work efficiently accomplish the purpose.

Requirement analysis can identify investigate, document, and analyse requirement of the system. Investigation step needs the system the significant activity of problem analysis. The role of system analyst determines the actual requirement of the system. System analyst possible makes a mistake in discover condition and try to analyse the problem. As a result, possible will have the wrong solution of design and implemented. The answer will not to solve the problem and cause a new challenge in the later system analysis process. Hence, this chapter will be discussing a fact-finding method for system analyst to collecting the information for improving the system of the project.

4.2 Fact-finding methods

Fact-finding is a process of collecting the data and information based on techniques, such as observation, questionnaires, and interviews. System analyst uses the appropriate fact-finding technique to develop and implement the system or project. Require of collecting facts are critical apply tools in SDLC (System Development Life Cycle), the reason because tools cannot be used effectively and efficiently without extracting from facts. The fact-finding method used in the earlier stage of SDLC including system analysis phase, design and post-implementation review. The facts were included any information system can have tested based on the steps, data-facts was used to make useful information, process-function to perform objectives and interface-design interact with users.

4.2.1 Observation

Observation aims to find out what happens, not what workers think observer expect to see. The point of observation is to see the people refer who does in the organisation and what information they do. By merely watching people to work, besides relatively straightforward observe how data flow around the system. Observation has to be careful of Hawthorne effect. Because is possible the users are modifying their behaviour, due to the reason they notice they are being watched. Furthermore, this is a useful technique for understanding the system. Is useful when the collected data was validity in question or when the complexity of specific aspect of system avoids clear description by end-user.

| Advantages | Disadvantages |
|--|--|
| Data collected can very reliable | Possible observe the wrong way |
| Be able to work measurements | Possible inconvenient of time |
| Inexpensive compared with other methods | Tasks not always performed the same way |
| Allow seeing accurately what being done in complex tasks | People possibly will perform differently when being observed |

Table 4. 1: Advantages and Disadvantages of Observation

4.2.2 Questionnaires

Questionnaires are prevalent method of fact-finding. A great deal of time was spent on preparing the questionnaires so that the right type of question will be asked and the most relevant information is collected. Moreover, the user fills the questions which given by system analyst and return the answers to system analyst. Questionnaires can

save time; this is because system analyst does not require to meet each user, and if the interview is short, questionnaires more useful. To fulfil requirements of system objective; system analyst should have ability define design and frame questionnaires. Questionnaires have two types,

4.2.2.1 Free-format questionnaire

A questionnaire designed provide respondent more considerable latitude in the answer. The question was asked, and respondent records answer in space provided after question.

4.2.2.2 Fixed-format questionnaire

Questionnaire are containing question require to select answer from predefined available responses. Example, Rating Question, Ranking Questions, and Multiple-Choice Questions.

| Advantages | Disadvantages |
|--|---|
| Allow for anonymity | Possible return rate often low |
| Answer the question quickly | Difficult prepare the question |
| Responses can be tabulated quickly | Cannot observe body language |
| Allow completing at people when free | No guarantee individual will answer all questions |
| An inexpensive way to collect data from a large number | No opportunity re-word or explain misunderstood questions |

Table 4. 2: Advantages and Disadvantages of Questionnaires.

4.2.3 Interviews

The interview is a most generally used technique to gather information from face-to-face interviews. Interviews purpose is to find, verify, clarify facts, motivate end-users involved, identify requirements and gather ideas and opinions. The analyst can formal, play politics, legalistic, or informal. Role of the interview includes interviews whom system analyst and interviewee who system, owner or user. As the success of an interview depends on the skill of analyst as interviewer. Interviewing technique require outstanding of communication skills for interaction between system analyst and user. The types of interview and the question is,

4.2.3.1 Unstructured Interview

This type is conducted question to answer session to obtain necessary information about the system.

4.2.3.2 Structured Interview

Standard questions which user need respond in either close to objective or open descriptive format.

4.2.3.3 Open-ended question

The question allow interviewee respond in any way.

4.2.3.4 Closed-ended question

The question restricts the answers to either specific choices or short, or direct responses.

| Advantages | Disadvantages |
|---|---|
| Allow analyst probe for more feedback | Time-consuming |
| Be able to observe nonverbal communication | Maybe impractical due location of interviewees |
| Provide analyst opportunity motivate interviewee to respond freely and openly | Success highly dependent on analyst human relation skills |

Table 4. 3: Advantages and Disadvantages of Interviews

4.2.4 The method of Fact-finding used in the Project (Questionnaire)

Based on three different methods of fact-finding, in this project are determined to use the questionnaire. This is because questionnaire is useful for the short-term of the project. Although the project is not long-term, using this fact-finding technique can be allowed to receive a large number of feedback. The survey can be posted on online and use other third-party web or software to do that, that also provides the feature like the statistical record to easier analysis the result wanted. Additionally, the method is cheaper than other methods and allow the people to answer the question when they free. In the end, the answer to the questionnaire can be used as an improvement of the system, such as design, the function of the system and added more feature on the project.

In this questionnaire, the survey will be decided to use the google form. This is because the online survey is more of saving cost and time to allow the customer or user can have completed filling the form at any free time. In this survey, the first section of the total question is 15. This 15 question has 14 requires the user to fill, and the last question is optional. The second section is only one optional question, is to ask the user to fill they email to us, as we can update the latest news and random the user to send the first digital copy of the game. The 15 question have 3 question is get the information of the user, 8 question is get the data about how user prefer and like in the game system, and the rest of 4 question is related in this project. The question about the information

of the user is gender, age, and occupation. The 8 question relate to how the user feels about the game system such as user are interested in playing the game, which type of game user is like, and which devices are preferred in a game. The 4 question relate on the project such as, which feature should approximately add in the game, the genre of the game is interested in the user, what feature are preferring in the game, and what the idea the game should look like.

As a result, have been displaying and what expected, the user is mainly male and teen or adult. This is because the majority of the gamer is male, they like to play the game to relax they stress compare female would like to watch the drama. Student and Full-time employee are also being primary target at the age group level. The user of answering the survey are mainly like to play the game on PC (Personal Computer), this is because of the highest spec and run the game more smoothly compared to the phone or portable video console. Every gamer will like to pay the game for they own, the majority of the user will pay the game around in USD 10, and prefer to buy the game in Digital Distribution Platform like STEAM. People will like to play a game of horror type, and the feature what he wants in apply to the game is First-Person View.

In short, the feature of all user wants to apply in the game are ensure the system will have made. The target user will be the teen or adult would like to play horror game for the journey. The majority is male but also have the possibility female will like to play the game and introduce to other friends as well. The game should good on gameplay; this is why the user is like to purchase and play the game. The whole questionnaire can determine and ensure what of the feature will complete added in the game, and what of the feature will add to improve the game performance. The questionnaire sample on **Appendices A**.

4.3 System requirement

A structured document is setting out detailed descriptions of system services. Something information system must do or property that must have. Also known as a business requirement. Business Requirement typically begins with a statement of specifying strategy or goal developed by the leadership team. Team identify business strategies and objectives to meet a business need arising from competitive, operational, regulatory and other pressures of business. System Requirement can be split into two

different requirements; Functional Requirement is something information system must do. Whereas, the Nonfunctional requirement is property or quality system must have, like Performance, Security, and Costs. The characteristics are, developed in parallel to Business Requirement, answer a question about solutions technical parameters required provide requested business functionality, and the combination of Business Requirements provide information needed for the project team to answer the question how to provide a solution and how many solutions will cost.

4.3.1 Functional Requirement

Statements services the system should provide, how the system should react particular inputs and how the system should behave in specify situation. The operation is equated with product or system features for which menu or button, like select item order, calculate the amount, and identify the customer. Based on the Project “You Are Not Alone”, the Functional Requirement is,

4.3.1.1 Button

The button will show it, Menu Page; this is requiring the user to Start Game, Continue Game, Exit Game, and Setting. The button function also possible work on Pause Menu. Each button will get a response to the specified action.

4.3.1.2 Movement

The user can use keyboard key (WASD) to move the character, alternative devices such as a controller (Left Stick) is work as a similar result. Crouch by using key (Ctrl) and controller (Left bump). Running allow the user to hold the input key (shift) and controller (Left Trigger). Regular Walking as the slow walk movement.

4.3.1.3 Camera View

The camera view function is allowing the user by using mice and controller (Right Stick). In short, this means the view of the game is allowing the user to use specify input devices to change the camera view. Another word, the camera view is the human eye in the game.

4.3.1.4 Action

The Action can be split into primary and secondary action. Primary action is like an open door, pick up an item, by using key ‘E’ and controller ‘A’ button. Secondary action such as interact with the items (note), by using key ‘T’ and Controller ‘Right Trigger’.

4.3.1.5 Flashlight

The game is involving two type of lighting sources, flashlight and phone flashlight. Is require user by pressing key 'F' or controller 'Y' to turn on or off the light, and switch between two sources by using the number key or mouse wheel and controller direction pad (D-pad up and down arrow). The important key is 'R' and controller button 'X'; this is because is require the user to reload the battery to get the power in full.

4.3.2 Nonfunctional Requirements

This requirement is constraints on services or function offered by a system such timing constraint, limitation on the development process, standard. Nonfunctional requirement refers the whole slew of attributes including security, performances levels, and various "ilities", like usability, reliability and availability.

4.3.2.1 Backup

The game will automate save the data when the user steps on to specify the location. The data had been stored in the game are saved into a file with .sav extension. The primary data including Player Spawn Location, Player health, the Player battery life of Flashlight and Phone, Player inventory item like painkiller pill, and battery amount.

4.3.2.2 Documentation

The game will provide brief documentation of tutorial for any novice user learn how to play the game. The first stage of the game mainly will display some text and descript to teach the user how to play the game. For example, how to walk, crouch, open door.

4.3.2.3 Readability

The script or the text is simple and undoubtedly to lead the user to understand quickly. For example, the tip of the tool in the game, like the user is close to one object can active, like Door. The tools tips will display a show text to remind the user can press the key like 'E' to open the door.

4.4 Conclusion

System Analysis is processed investigation of system operation intending changing to new requirements or improving current working. The fact-finding method allows the

developer and design to use the data gathered from the people; they provide the answer be helpful to increase the system performance.

Requirements set out what system should do and define constraints on operation and implementation. Functional requirements set out services that system should provide, Non-functional requirement restrict system being developed or development process.

This chapter mainly used to find out what requirement of a system for the designer, the application meets the customer needs. Finding out the requirement for the system be able to help the system and improved. The requirement is essential to create successful software project as will help design system which fulfils the requirement.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

In this chapter, to understand more about the overview Modeling and several diagrams will be discussing, the sketch of the User Interface Design and System Design will be explaining briefly. The picture is worth thousand words, the reason why the diagram was created (UML). Unified Modeling Language to forge common visual language in the complex world of software development that understandable for anyone who wants to understand the system. The first of the design User Interface will be display on this chapter, and summaries of discussing what is the use of.

5.2 Modelling

Modelling is about built a representation of things in the world and allow ideal to investigate, is central to all activities in the process for making or create an artefact of some form. Model is a way to express a particular view of the identifiable system. From the Model, to allow understanding, the problem involves in building something. The Modelling also aid to communicate between involved in the project, especially between the user and part of interested people. Model Is abstraction, which allows people concentrate on essential of the problem by keeping out non-essential details. Due there is a limit about how many people can understand at one time. To build the model is mainly help in activities such development of large software systems. For example, the developer makes several models throughout the development process to verify final software system will meet the requirements.

Modelling also call as UML (Unified Modeling Language). UML as Modeling was created to provide more natural view and ideal for interested people want to

understand about the system, reduce and avoid the complicated information for people. UML is analogous to blueprints used in other fields and consists of different types of diagrams. UML diagrams describe the boundary, structure, and behaviour of a system and the objects within.

5.2.1 Data Flow Diagram

DFD (Data Flow Diagram) maps the outflow of information for any system and process. DFD is used to define a symbol like rectangles, circles, and arrows with label text to show the input data, output data, where the storage point and the routes between each destination. Data flowcharts can range from simple, hand-drawn process overviews, and in-depth multi-level DFDs for displaying progressively deeper details of how data is handled. DFDs, like any other diagram, can often visually display things that would be difficult to explain in words and work for both technical and nontechnical audiences, from developers to CEOs.

The reason for using DFD is because it graphically represents a function or process which captures, manipulates, stores, and distributes data between a system and its environment and between components of the system. Visual representation makes useful communication between users and developers. The structure of DFD allows starting from a broad overview and expanding to a hierarchy of detailed diagrams. Several points of the reason are the simplicity of notation and the provision of logical information flow of the system, also determining physical system construction requirements and establishing the manual.

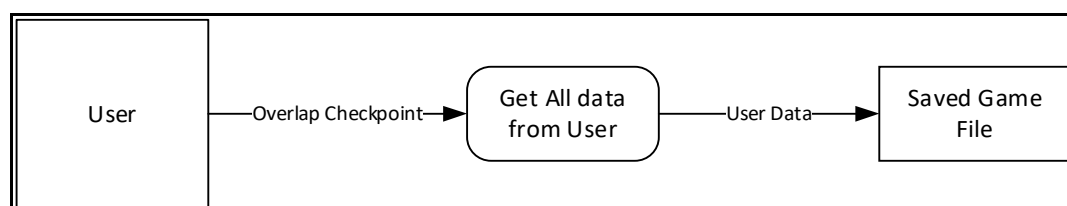


Figure 5. 1: Data Flow Diagram

From Figure 5.1, the diagram shows how the project can save user data. When a player steps on the box collision, the overlapped event will be triggered. Take all the information of the data from the user and pass it to the Save Game object to save the data in the file.

5.2.2 Flowchart

The flowchart is diagram depicts process, system or computer algorithm. This diagram is widely used in multiple fields to document, study, plan, improve and communicate complicated process in the clear, straightforward understand diagram. Flowcharts, use rectangles, ovals, diamonds and potentially numerous other shapes define the type of step, along with connecting arrow determine flow and sequence. This diagram can range from simple, by hand-drawn charts to complete computer drawn diagrams depicting multiple steps and routes. If consider all various form of flowcharts, this diagram is most simple diagrams on the planet, used by both technical and non-technical people in numerous field. Flowchart sometime called by Functional Flowchart, Business Process Mapping, BPMN (Business Process Modeling and Notation), or PFD (Process Flow Diagram). The diagram is relating another famous diagram such Data Flow Diagram at above 5.2.1, and UML Activity Diagram at below 5.2.4.

The flowchart is a visual representation of a sequence of steps and decision needed to perform the process. Each step in sequence noted within diagram shape. Step are linked by connecting line and directional arrow; this allows anyone view flowchart and logically follow the process from beginning to end.

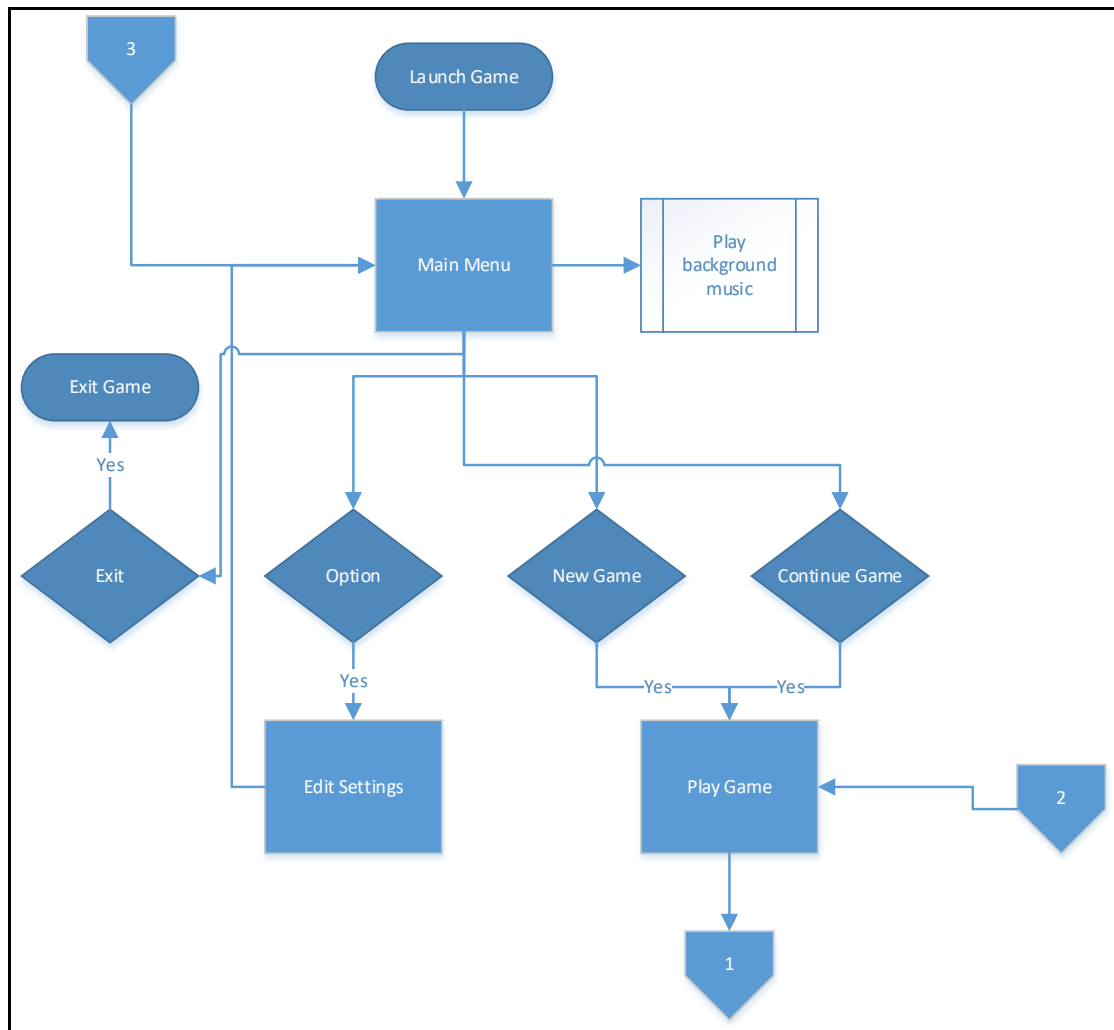


Figure 5. 2: Flowchart (page 1)

In Figure 5.2, the first flowchart can see the process relate to the Main Menu. When Player launch the game, the next process will move toward into Main Menu Screen. On the Main Menu will have subprocess such playing the background music and several buttons to process the next section. The First and Second button is relating to Game, such Continue Game Button and New Game Button. This button will start the gameplay. Options button will proceed to Edit Settings Screen When Done or Cancel also will bring back to Main Menu. The last button is Exit if user press this button will Exit the Game.

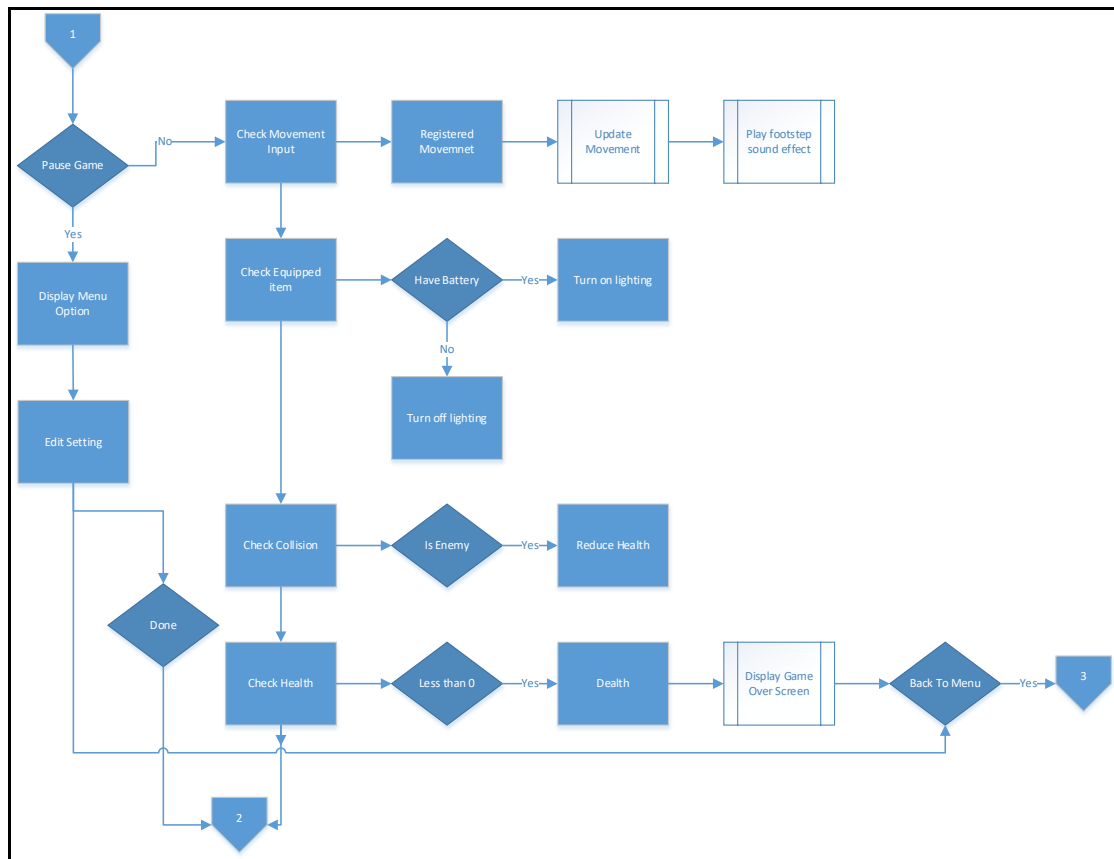


Figure 5. 3: Flowchart (page 2)

In Figure 5.3, the continuous Flowchart of 5.2. When the Game is playing, if user press Pause Game, will display the Pause Menu Screen and allow the user to alter the option. If Done or Cancel will continue the gameplay however if user press Back to Menu button, will bring the user back to Main Menu Screen. Else, The System Will Check the Movement input, registered the movement of the character, with animation movement and footstep sound effect. After that, Check the player equipment, if the battery is having, the item can turn on the light else not. Furthermore, if the player is overlap certain collision such as Enemy, will be damaged. If the life of a player is less than 0, will end the game, meaning that Game Over Screen will display and allow the user to Back to Menu.

5.2.3 Use Case Diagram

In UML, user case diagram can summaries the details of system users, also known as actors and the user interaction with the system. To build the Use Case Diagram is required to use a set of specialised symbols and connectors. An efficient Use Case

Diagram is able to help the developers discuss and represent the scenarios which system interact with people, what is the goals to help the entities (Actors) to achieve and the scope of the system.

UML use case diagram is primary form system or software requirement for new software program underdeveloped. Use Case specify expected behaviour and not exact method making happened. Use case once defined can denoted both textual and visual representation. Fundamental concept of this modelling is help developer design system from an end user perspective, is a useful technique for communicating system behaviour in user term by specifying all externally visible system behaviour.

The purpose of Use Case Diagram is dynamic capture aspect of the system. However, the definition is too generic describe the purpose as another diagram such as activity. With the same purpose. Use Case Diagram used to gather requirement of a system including internal and external influences. To specify the context of the system, validate system architecture, drive implementation and generate test case and developed by analysts together with domain experts.

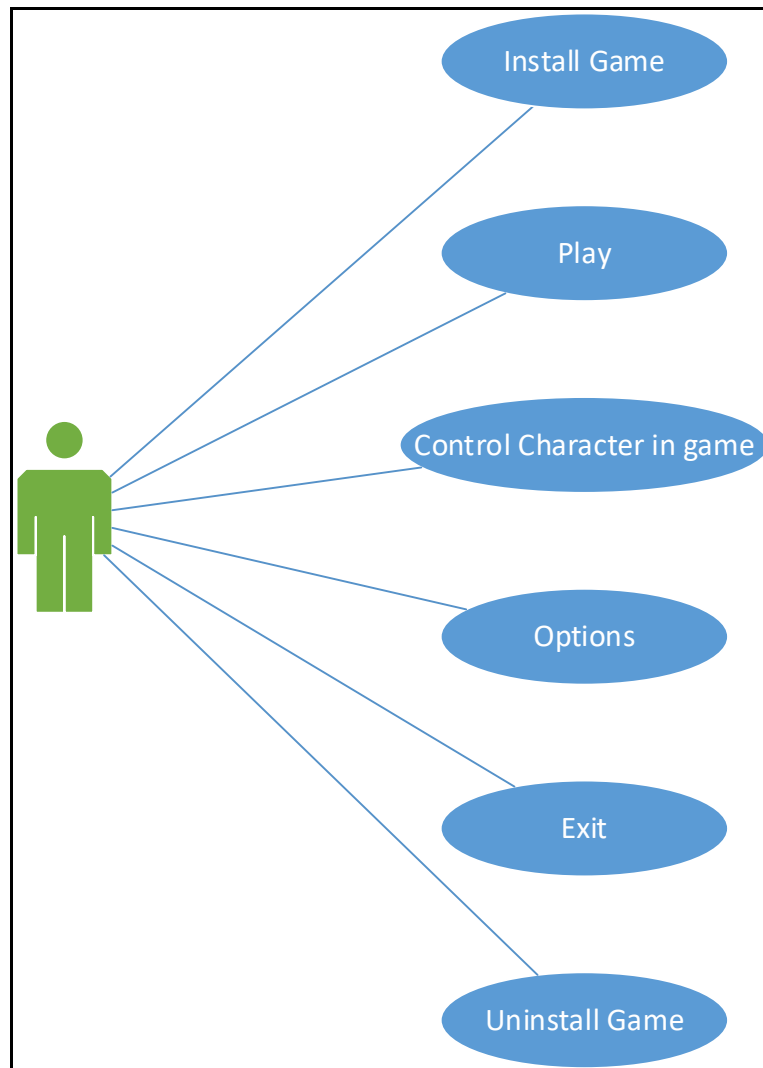


Figure 5. 4: Use Case Diagram

In Figure 5.4, to display what are the activity user mainly can do. Player allows to Install and Uninstall the Game. Play the game with Start new level or Continue Last Played. Player is able to Control the Character in the game, to achieve the story and finish. Player allows to alter the setting, and last activity is Exit the game once finish or want to take the rest.

5.2.4 Activity Diagram

UML includes several subset diagrams, including behaviour diagram, structure diagram and interaction diagram. Activity diagram along with Use Case and State Machine Diagram are considered behaviour diagrams because the diagram describes what must happen in system being model.

Stakeholders have many issues to manage, essential to communicate with clarity and brevity. Activity diagram helps people on business and development sides of the organization come together to understand same process and behaviour. Used set of specialised symbols including used for starting, ending, merging, or receiving step in flow to make activity.

Primary purposes of activity diagram are similar other four diagram, such as Use Case Diagram. This diagram are captured the dynamic behaviour of the system. Other four diagram is used to show message flow from one object to another, however, activity diagram used to indicate message flow from one activity to another.

Activity is the particular operation of the system. Activity diagram not only used for visualising dynamic nature of system but also used to construct an executable system by using forward and reverse engineering techniques. Only message part is missing in this diagram. The message does not show in this activity to another, activity diagram sometimes considers ad flowchart from 5.2.2. Although diagram looks like a flowchart, different flows such parallel, branched, concurrent and single. The primary purpose of this diagram can describe as draw activity flow of the system, describe sequence from activity to another and parallel, branched and concurrent flow of the system.

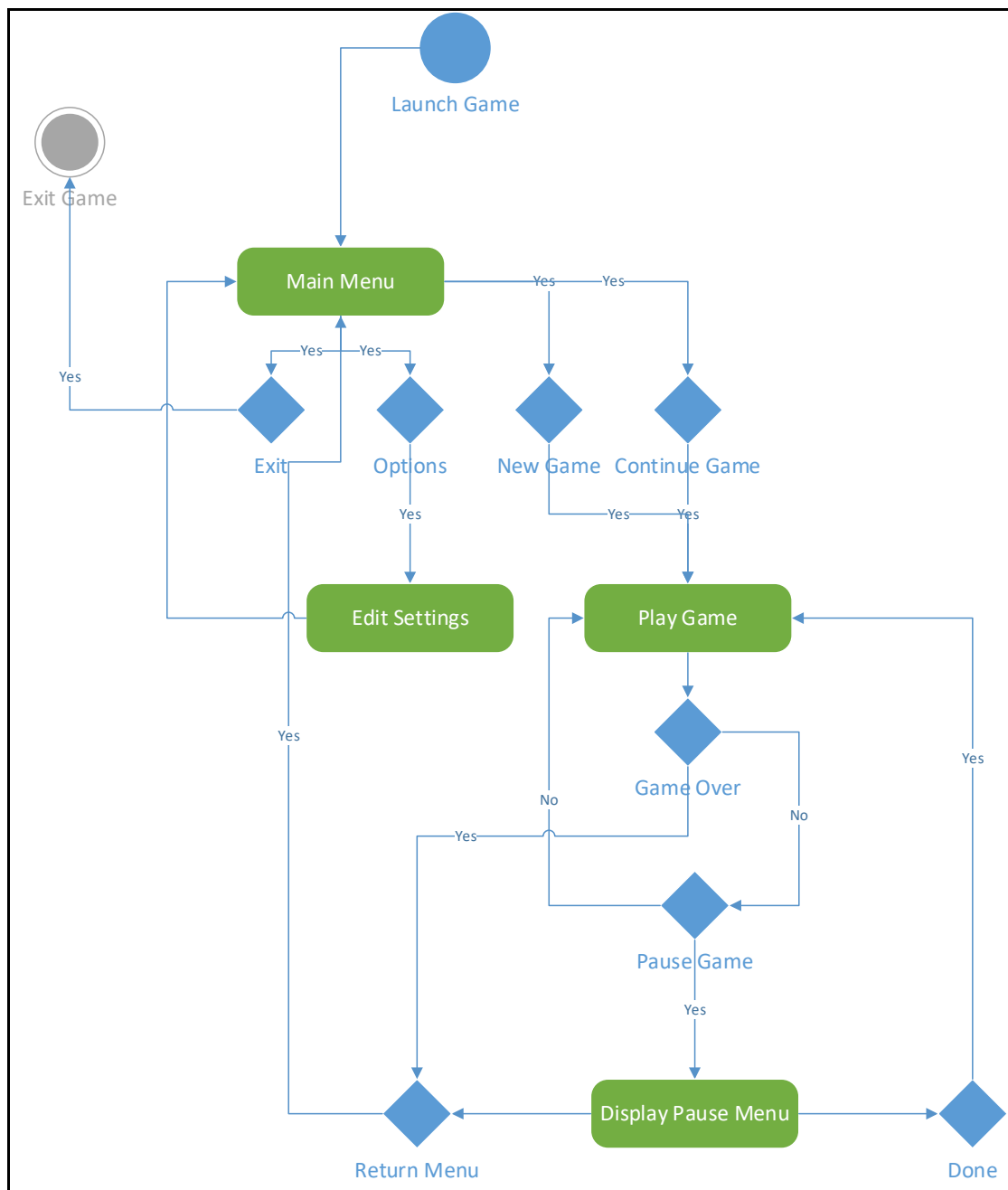


Figure 5. 5: Activity Diagram

On Figure 5.5, is the Activity Diagram briefly about how the game process. This diagram is similar to the Flowcharts, 5.2.2. However, the differences is not design in detail in the Flowcharts process. On this diagram, user launches the game and send to the Main Menu. On Main Menu Screen have four central buttons. Continue Game and Start Game are proceeding to Play Game process. Option button allows the user to alter the setting and Exit Button to Exit the application. When User is playing the game, if the game is finish or Game Over, the game will allow the user to return the Main Menu. If not, and the User is Pause the Game, the Pause Menu will be displayed and after all

the actors have been done, like modify setting or pause for rest. When Done the process will continue run the game else will send to the Menu Screen.

5.3 User interface design

In this section, will show the brief idea of the User Interface Design on the application, may possible different on the real application.

5.3.1 Main Menu User Interface

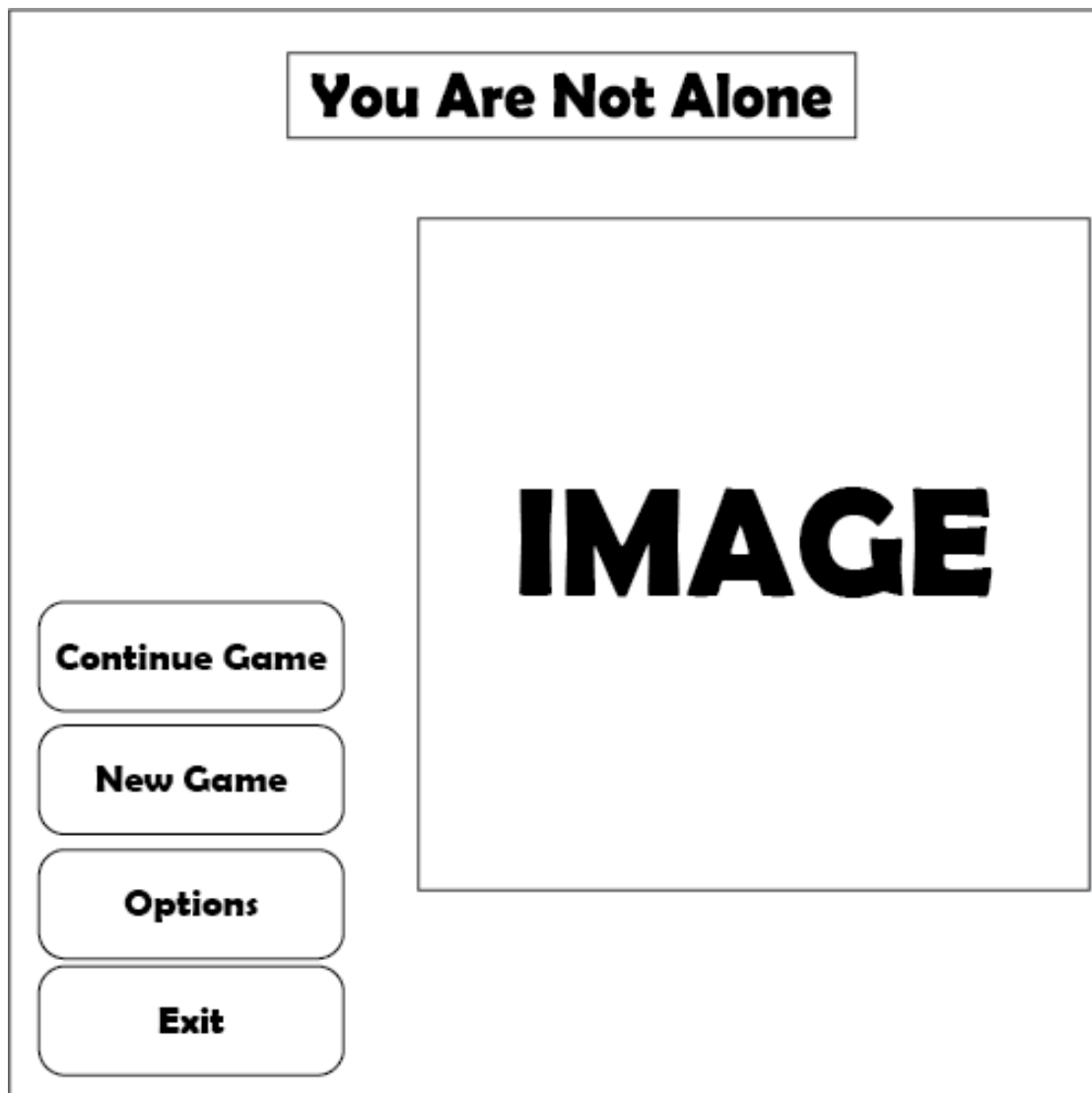


Figure 5. 6: Main Menu UI

On Figure 5.6, are the brief idea of the Main Menu Screen. The top will be the Heading of this application title. The title should Bold and Bigger font to lead user know what is the application they were playing. On the Left-hand side will have the Main icon of the Image to represent the game. The Image as the background Image as well. Next, on

right-hand side will have four buttons. Continue Game, New Game, Options, and Exit. Continue Game will load the last played data of user playing, New Game will start the game from begin, Options button allows the user to change the setting such as Resolution, and Last Exit button is allowing the user to Exit the Game.

5.3.2 Pause Menu User Interface

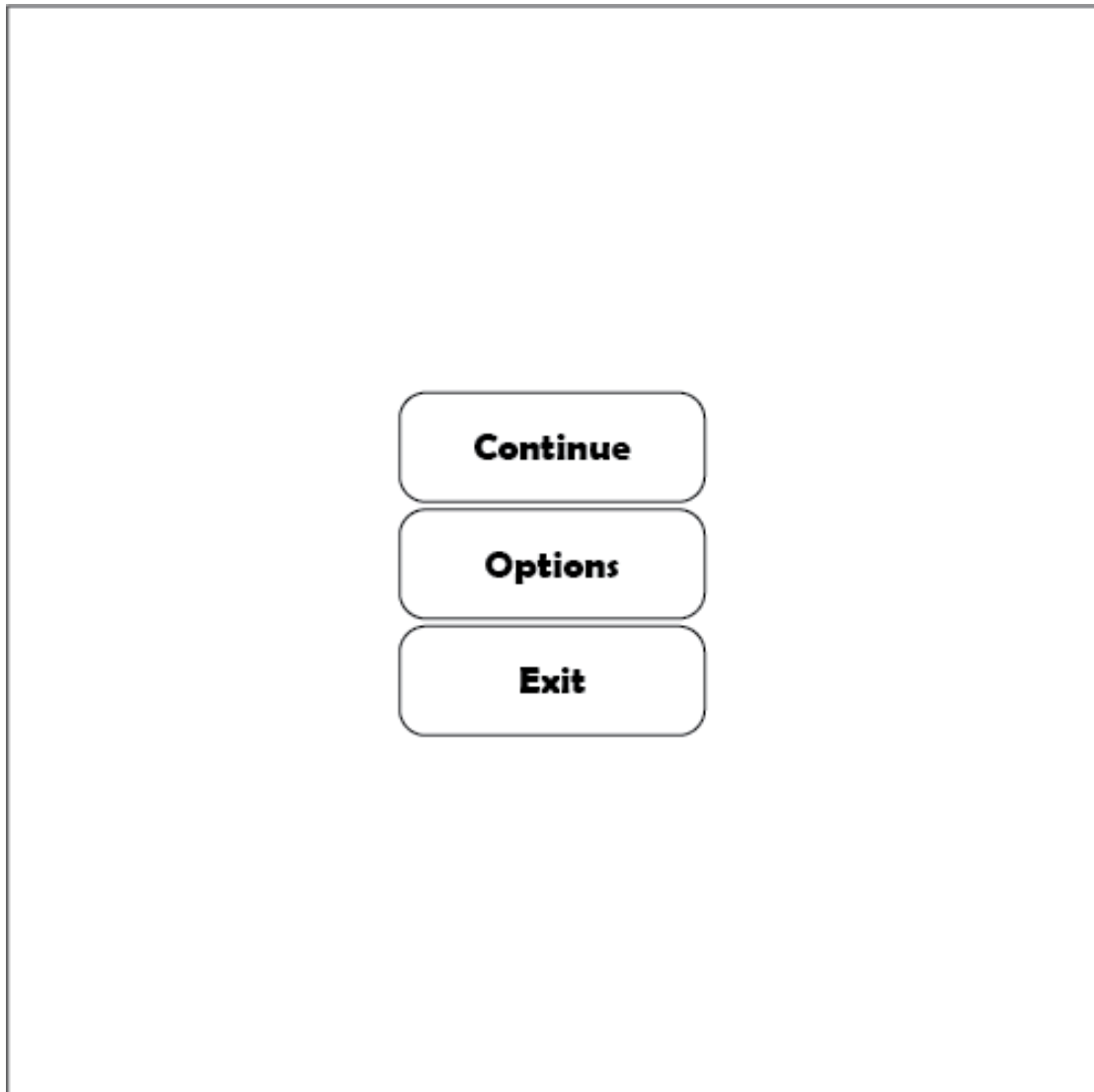


Figure 5. 7: Pause Menu UI

On Figure 5.7, will display the Pause Menu. When the player is playing the game, however, need to take a rest or alter some setting. This Pause Menu will display after that pressed the key to Pause. In this menu will have three central buttons. Continue button mean Resume the game. Option Button will display other Menu to the able user to modify the setting. Exit button will return to Main Menu Screen.

5.3.3 Options Menu User Interface

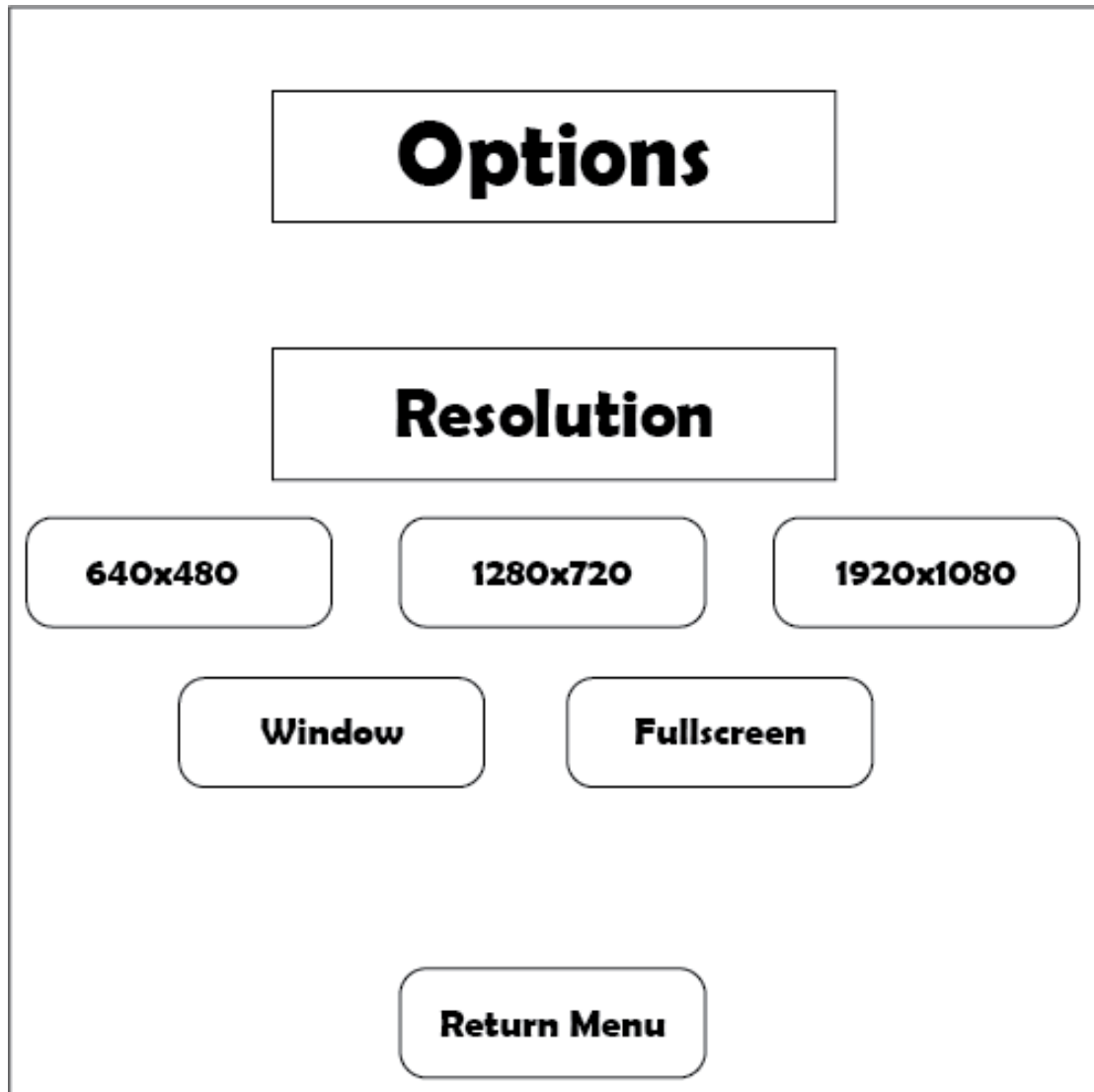


Figure 5. 8: Options Menu UI

In Figure 5.8, this Menu will be displayed after the user clicked the Options button for modifying the setting of the game. The Main setting needs to be changed in the game will be Resolution. On the top centre side will have the title of this menu. After that will have the heading of this option. Next three buttons will allow user alter the resolution of this game. Such as 640x480. The other two button is allowing user set the game to Full screen or Window form. The last button on the bottom centre side is Return the Menu, this possible return to Main Menu when the user is clicked from the Main Menu (figure 5.6) or Pause Menu (figure 5.7) when the user clicked from the Pause Menu.

5.3.4 Buildings

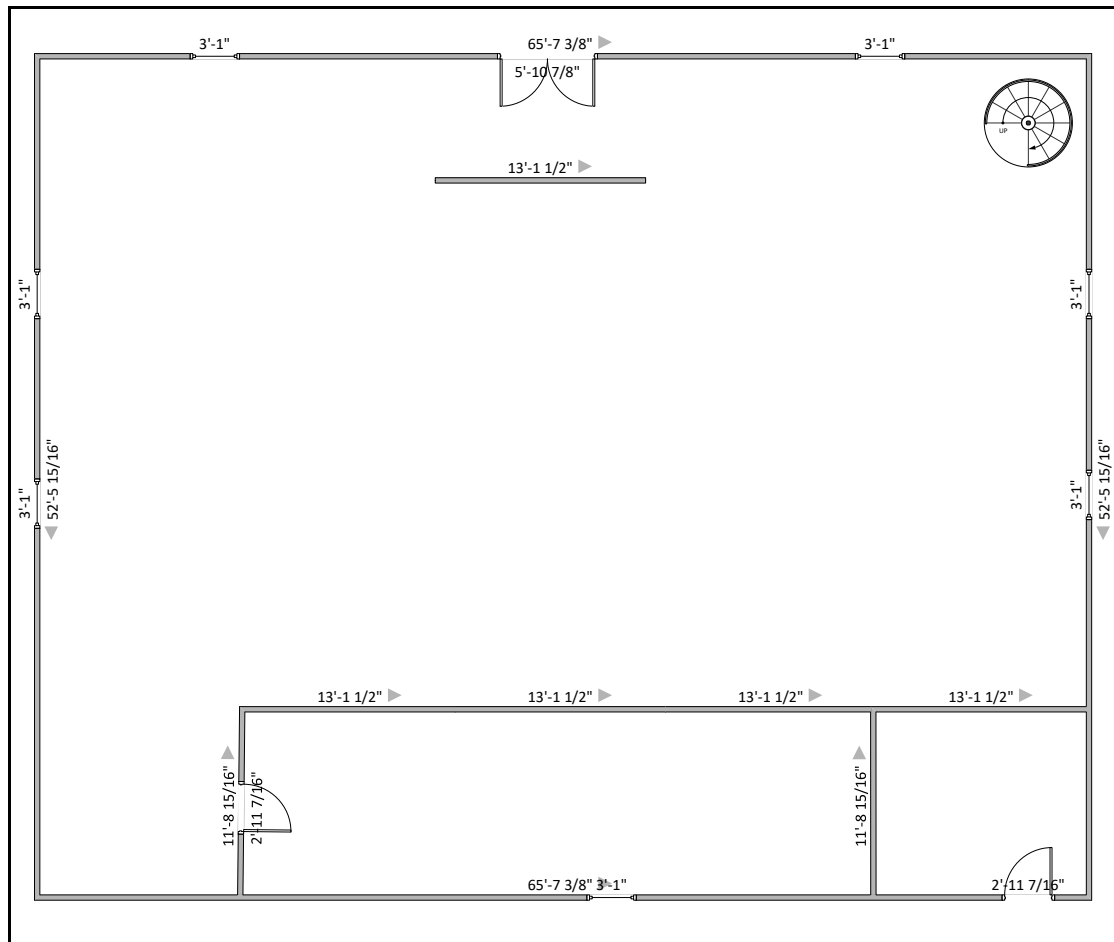


Figure 5. 9: First Building (Abandoned Motel, first floor)

On the figure above, 5.9. Is the first building inside the game. The building is an abandoned Motel. This Motel like ordinary house, only the front of the wall will allow the user to registered such check in and check out process, and backside has Kitchen and Store Room. On the leading place have a spiral stair to move to the second floor.

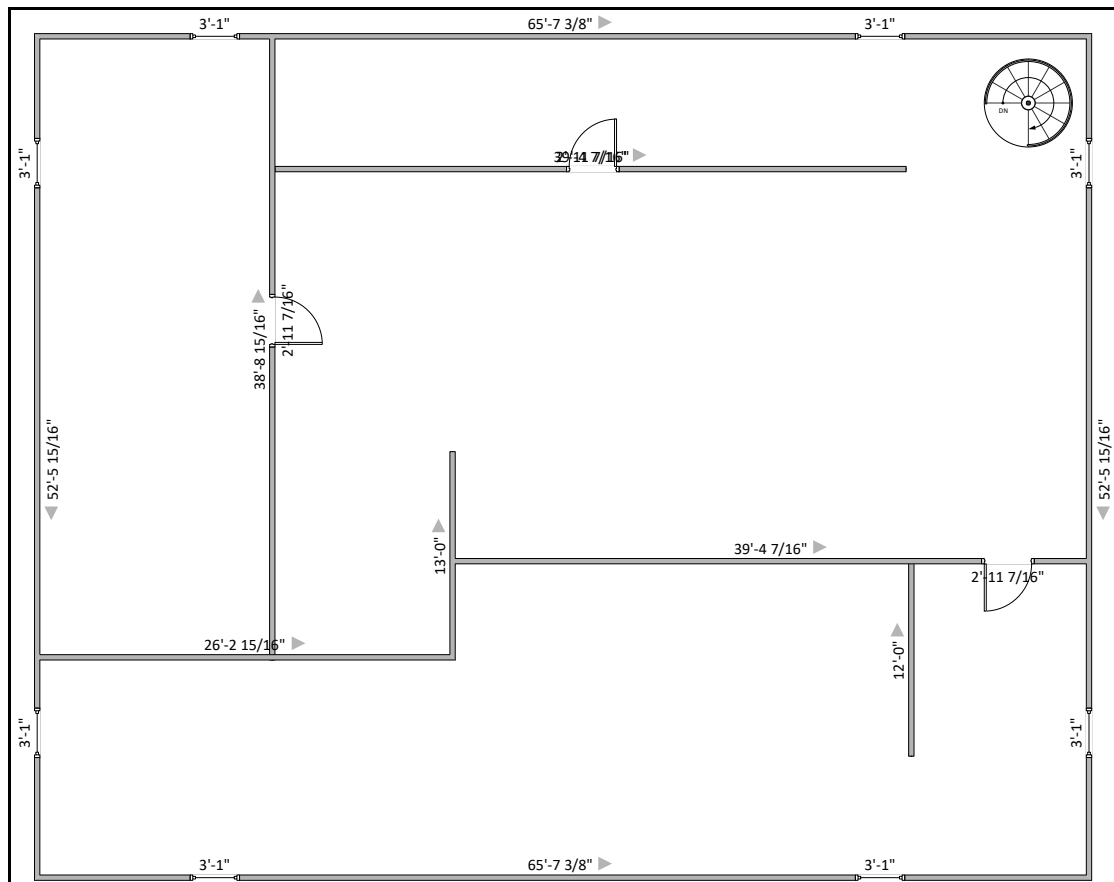


Figure 5. 10: First Building (Abandoned Motel, second floor)

In Figure 5.10, the second floor of the Abandoned Motel. This floor most will allow the customer to rent and use. Will have several rooms, each room will have different furniture and surprise in the game. The front of left side will have the stair move to the first floor of figure 5.9.

5.4 System structure

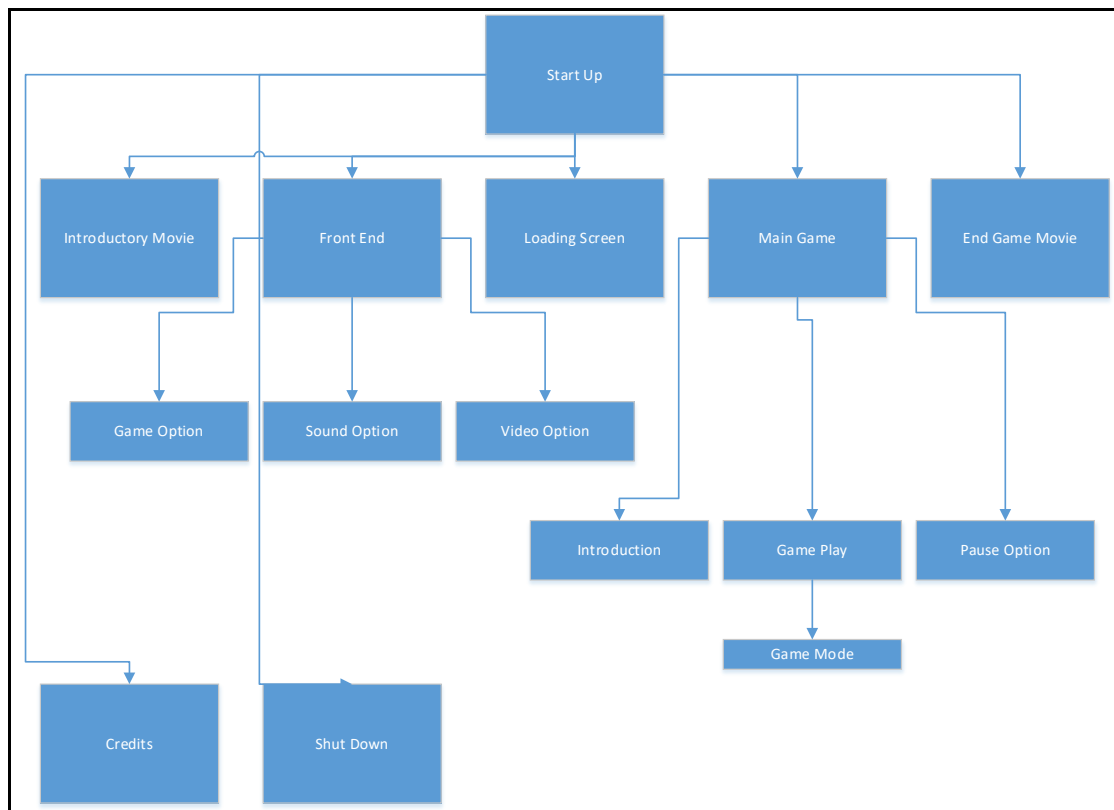


Figure 5. 11: System Structure

In Figure 5.11, the overall of the System Structure. When User Start Up the game, will display the Introductory Movie, such as which Game Engine is using or who are the sponsor or supporter. On the Front End, that has several options available be able the user can change, such as Game Option, Sound Option and Video Option. This option will have based on user preferences and present. When the game Start or load, sure will have the Loading Screen. Loading Screen to notify the user of the game data are loaded, please take a short time to wait, instead of not response effect. On the Main Game menu, will have three main actions, Such as Introduction the whole game, for example like a prologue. How the Game Play, the gameplay will have the own style of Game Mode, the game of the playstyle While playing the game allow the user to Pause this is because can enable the user to take rest or adjust the setting option. When the user is completed the game will display the End Game Movie, also the Credit. In the end, user allows to Shut Down the Game any time they prefer.

5.5 Conclusion

Providing system architects, software engineers, and software developers with tools for analysis, design and implementation of software-based system for modelling business and similar processes. Advancing state of the industry by enabling object visual modelling tool interoperability. However, allow meaningful exchange model information between tools, agreement on semantics and notation is required.

In saying that, application allows customer and prospect to learn more about the products offering by encouraging dialogue through effective communication and simplifying the end goals. Usability is the foundation of the successful user interface and most accessible process for creating positive user experiences. Positive user experience is a critical part of overall customer experiment and helps define the relationship with customer favour.

In simple term, Modelling allows the developer and interested people to understand how the system structure and the flow of the application. People are non-technical also understand the primary structure of the system. Simple and easier. User Interface allows the developer create a fundamental idea of design in the game; this requires the change of customer need and satisfaction.

CHAPTER 6

IMPLEMENTATION AND PROGRAMMING SYSTEM

6.1 Introduction

In this chapter will discuss the module implementation, part of the application which coding part. Provide part of a screenshot of coding and briefly explain what is that mean, and provide several screenshots of the system. How will the game work in real time? The more coding screenshot will provide in the section **Appendices B**.

6.2 Module implementation

This section will split into two subsection, one is explaining part of code function, and other one provide the gameplay image. Develop this system have implemented two different ways, one is native C++ code, and other is Unreal Engine feature, Blueprint. In Unreal, generally will need to use both to get the better performance.

C++ is native code to use in Unreal Engine, the main things to use C++ is performance is well, however, specific function does not work nicely such animation not allow to do ease in and out effect, mostly will continuously. Blueprint is perfect for debugging purpose, during gameplay that time you still can view your code to understand which node is running or fail. The animation is another option will be using blueprint however performance and optimisation are not good than C++.

6.2.1 System Screenshot (Code)

6.2.1.1 C++

```
#pragma once

#include "CoreMinimal.h"
#include "GameFramework/Character.h"
#include "YouAreNotAloneCharacter.generated.h"

UCLASS(config=Game)
class AYouAreNotAloneCharacter : public ACharacter
{
    GENERATED_BODY()

    /** Camera boom positioning the camera behind the character */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
    class USpringArmComponent* CameraBoom;

    /** Follow camera */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category = Camera, meta = (AllowPrivateAccess = "true"))
    class UCameraComponent* FollowCamera;
public:
    AYouAreNotAloneCharacter();

    /** Base turn rate, in deg/sec. Other scaling may affect final turn rate. */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
    float BaseTurnRate;

    /** Base look up/down rate, in deg/sec. Other scaling may affect final rate. */
    UPROPERTY(VisibleAnywhere, BlueprintReadOnly, Category=Camera)
    float BaseLookUpRate;

protected:
    /** Resets HMD orientation in VR. */
    void OnResetVR();

    /** Called for forwards/backward input */
    void MoveForward(float Value);

    /** Called for side to side input */
    void MoveRight(float Value);

    /**
     * Called via input to turn at a given rate.
     * @param Rate This is a normalized rate, i.e. 1.0 means 100% of desired turn rate
     */
    void TurnAtRate(float Rate);

    /**
     * Called via input to turn look up/down at a given rate.
     * @param Rate This is a normalized rate, i.e. 1.0 means 100% of desired turn rate
     */
    void LookUpAtRate(float Rate);

    /** Handler for when a touch input begins. */
    void TouchStarted(ETouchIndex::Type FingerIndex, FVector Location);

    /** Handler for when a touch input stops. */
    void TouchStopped(ETouchIndex::Type FingerIndex, FVector Location);

protected:
    // APawn interface
    virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;
    // End of APawn interface

public:
    /** Returns CameraBoom subobject **/
    FORCEINLINE class USpringArmComponent* GetCameraBoom() const { return CameraBoom; }
    /** Returns FollowCamera subobject **/
    FORCEINLINE class UCameraComponent* GetFollowCamera() const { return FollowCamera; }
};
```

Figure 6. 1: YouAreNotAloneCharacter Class


```

#include "YouAreNotAloneCharacter.h"
#include "HeadMountedDisplayFunctionLibrary.h"
#include "Camera/CameraComponent.h"
#include "Components/CapsuleComponent.h"
#include "Components/InputComponent.h"
#include "GameFramework/CharacterMovementComponent.h"
#include "GameFramework/Controller.h"
#include "GameFramework/SpringArmComponent.h"

// AYouAreNotAloneCharacter

AYouAreNotAloneCharacter::AYouAreNotAloneCharacter()
{
    // Set size for collision capsule
    GetCapsuleComponent()->InitCapsuleSize(42.f, 96.0f);

    // set our turn rates for input
    BaseTurnRate = 45.f;
    BaseLookUpRate = 45.f;

    // Don't rotate when the controller rotates. Let that just affect the camera.
    bUseControllerRotationPitch = false;
    bUseControllerRotationYaw = false;
    bUseControllerRotationRoll = false;

    // Configure character movement
    GetCharacterMovement()->bOrientRotationToMovement = true; // Character moves in the direction of input...
    GetCharacterMovement()->RotationRate = FRotator(0.0f, 540.0f, 0.0f); // ...at this rotation rate
    GetCharacterMovement()->JumpZVelocity = 600.f;
    GetCharacterMovement()->AirControl = 0.2f;

    // Create a camera boom (pulls in towards the player if there is a collision)
    CameraBoom = CreateDefaultSubobject<USpringArmComponent>(TEXT("CameraBoom"));
    CameraBoom->SetupAttachment(RootComponent);
    CameraBoom->TargetArmLength = 300.0f; // The camera follows at this distance behind the character
    CameraBoom->bUsePawnControlRotation = true; // Rotate the arm based on the controller

    // Create a follow camera
    FollowCamera = CreateDefaultSubobject<UCameraComponent>(TEXT("FollowCamera"));
    FollowCamera->SetupAttachment(CameraBoom, USpringArmComponent::SocketName); // Attach the camera to the end of the
    FollowCamera->bUsePawnControlRotation = false; // Camera does not rotate relative to arm

    // Note: The skeletal mesh and anim blueprint references on the Mesh component (inherited from Character)
    // are set in the derived blueprint asset named MyCharacter (to avoid direct content references in C++)
}

// Input

void AYouAreNotAloneCharacter::SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent)
{
    // Set up gameplay key bindings
    check(PlayerInputComponent);
    PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &ACharacter::Jump);
    PlayerInputComponent->BindAction("Jump", IE_Released, this, &ACharacter::StopJumping);

    PlayerInputComponent->BindAxis("MoveForward", this, &AYouAreNotAloneCharacter::MoveForward);
    PlayerInputComponent->BindAxis("MoveRight", this, &AYouAreNotAloneCharacter::MoveRight);

    // We have 2 versions of the rotation bindings to handle different kinds of devices differently
    // "turn" handles devices that provide an absolute delta, such as a mouse.
    // "turnrate" is for devices that we choose to treat as a rate of change, such as an analog joystick
    PlayerInputComponent->BindAxis("Turn", this, &APawn::AddControllerYawInput);
    PlayerInputComponent->BindAxis("TurnRate", this, &AYouAreNotAloneCharacter::TurnAtRate);
    PlayerInputComponent->BindAxis("LookUp", this, &APawn::AddControllerPitchInput);
    PlayerInputComponent->BindAxis("LookUpRate", this, &AYouAreNotAloneCharacter::LookUpAtRate);

    // handle touch devices
    PlayerInputComponent->BindTouch(IE_Pressed, this, &AYouAreNotAloneCharacter::TouchStarted);
    PlayerInputComponent->BindTouch(IE_Released, this, &AYouAreNotAloneCharacter::TouchStopped);

    // VR headset functionality
    PlayerInputComponent->BindAction("ResetVR", IE_Pressed, this, &AYouAreNotAloneCharacter::OnResetVR);
}

```

Figure 6. 2: YouAreNotAloneCharacter Class .cpp (Part 1)

```

void AYouAreNotAloneCharacter::OnResetVR()
{
    UHeadMountedDisplayFunctionLibrary::ResetOrientationAndPosition();
}

void AYouAreNotAloneCharacter::TouchStarted(ETouchIndex::Type FingerIndex, FVector Location)
{
    Jump();
}

void AYouAreNotAloneCharacter::TouchStopped(ETouchIndex::Type FingerIndex, FVector Location)
{
    StopJumping();
}

void AYouAreNotAloneCharacter::TurnAtRate(float Rate)
{
    // calculate delta for this frame from the rate information
    AddControllerYawInput(Rate * BaseTurnRate * GetWorld()->GetDeltaSeconds());
}

void AYouAreNotAloneCharacter::LookUpAtRate(float Rate)
{
    // calculate delta for this frame from the rate information
    AddControllerPitchInput(Rate * BaseLookUpRate * GetWorld()->GetDeltaSeconds());
}

void AYouAreNotAloneCharacter::MoveForward(float Value)
{
    if ((Controller != NULL) && (Value != 0.0f))
    {
        // find out which way is forward
        const FRotator Rotation = Controller->GetControlRotation();
        const FRotator YawRotation(0, Rotation.Yaw, 0);

        // get forward vector
        const FVector Direction = FRotationMatrix(YawRotation).GetUnitAxis(EAxis::X);
        AddMovementInput(Direction, Value);
    }
}

void AYouAreNotAloneCharacter::MoveRight(float Value)
{
    if ( (Controller != NULL) && (Value != 0.0f) )
    {
        // find out which way is right
        const FRotator Rotation = Controller->GetControlRotation();
        const FRotator YawRotation(0, Rotation.Yaw, 0);

        // get right vector
        const FVector Direction = FRotationMatrix(YawRotation).GetUnitAxis(EAxis::Y);
        // add movement in that direction
        AddMovementInput(Direction, Value);
    }
}

```

Figure 6. 3: YouAreNotAloneCharacter Class .cpp (Part 2)

This .h class will declare all the variable will be used in later. PROPERTY variables declared using C++ syntax with additional descriptors, such variable specifiers and metadata placed above declaration. Different properties have a different function. For example, VisibleAnywhere is to Indicates this property is visible in all property

windows, but cannot edit. `BlueprintReadOnly` is property can read by Blueprints, but not modified. The category is Specifies the category of the property when displayed in Blueprint editing tools. `AllowPrivateAccess` is to defined in C++ private scope will be accessible to blueprints.

In overall, this is the class for the primary based on the character. Without this class, Character is not able to control and perform an action.

6.2.1.2 Blueprint

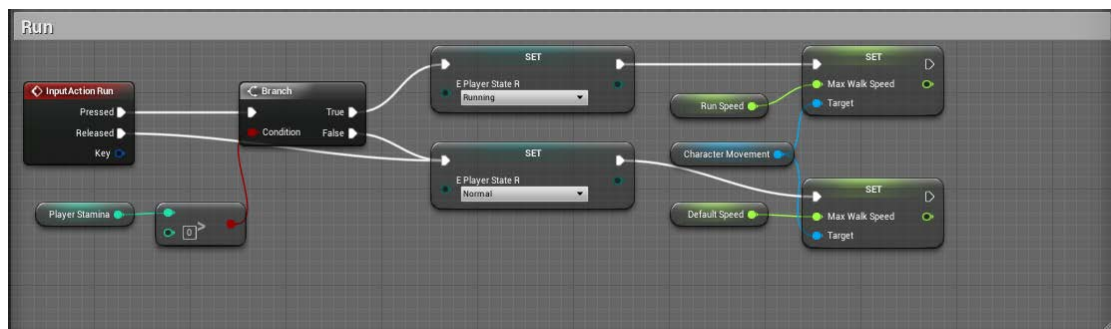


Figure 6. 4: BP Player (Running Event)

From the figure 6.4 is the event to trigger the user for running. If the player available the stamina, is allow to activate Running mode or else will still in the current mode as Normal stats.

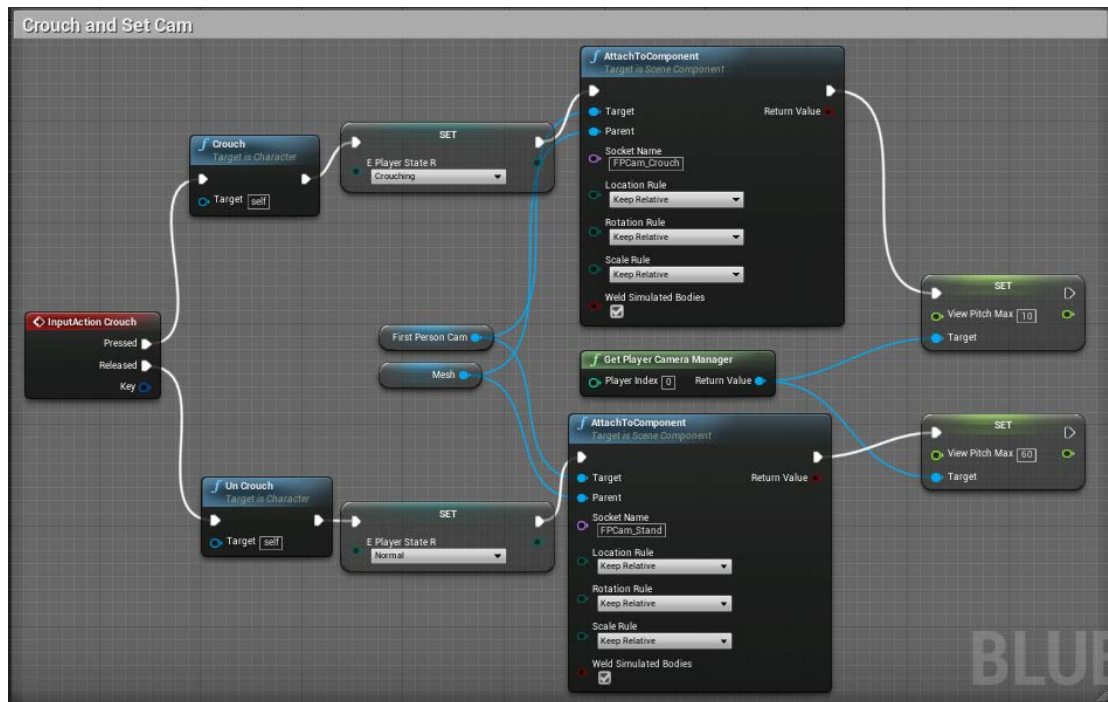


Figure 6. 5: BP Player (Crouch Event)

From the Image 6.5, is the event to trigger the user to crouching and set the camera restriction avoid user to control unrealistic angle.

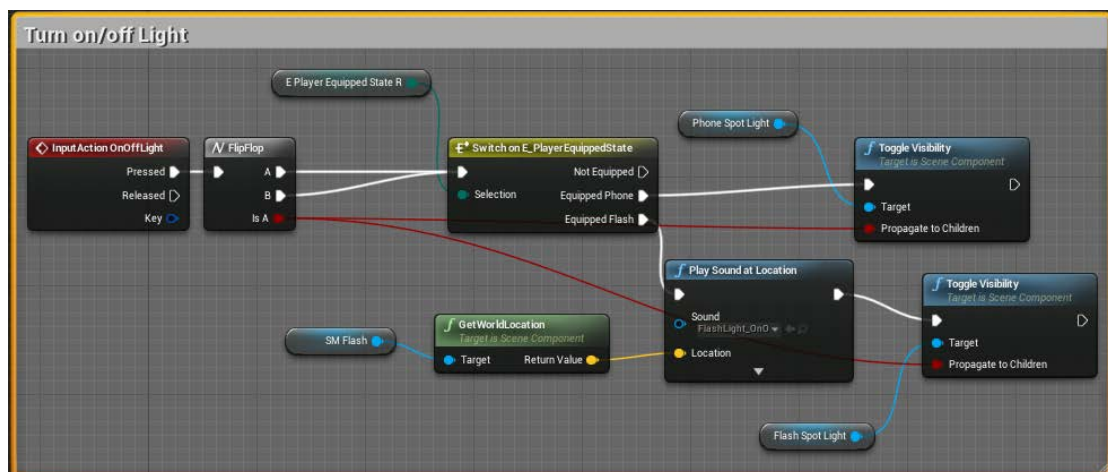


Figure 6. 6: BP Player (On/Off Light Event)

Figure 6.6 provides the method how the user allow to toggle light source from different equipment.

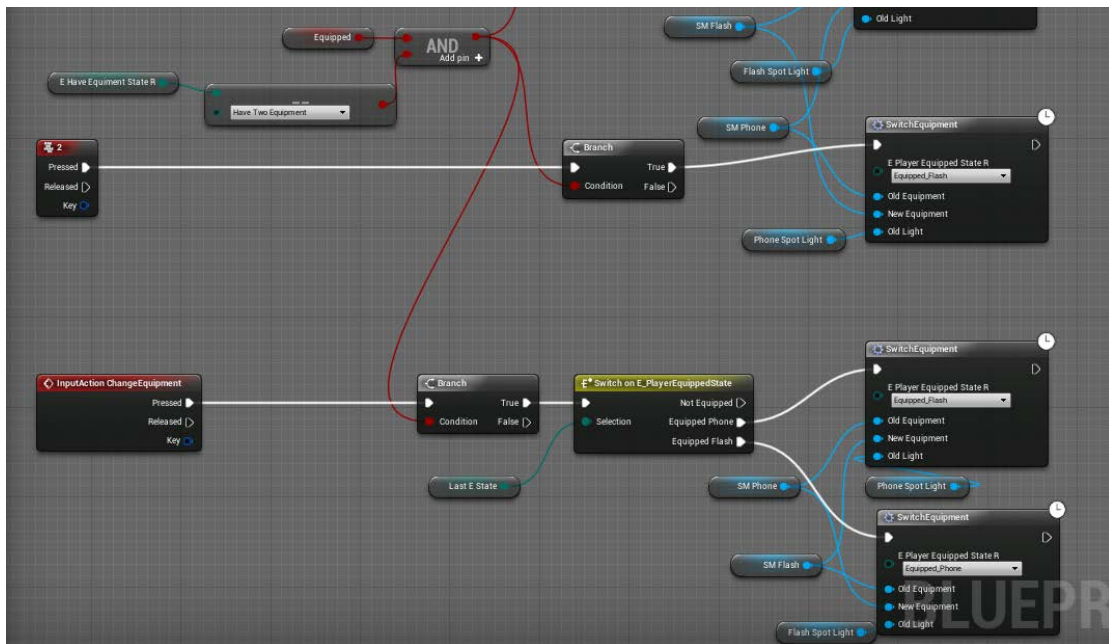


Figure 6. 7: BP Player (Switch Event)

Image above is present how the user able to switch the equipment from phone to flashlight, vice versa. By using Mouse Wheel on the keyboard or D-pad up and down arrow key.

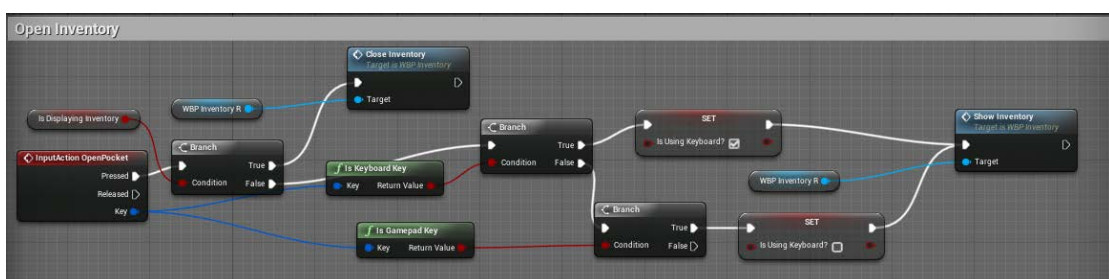


Figure 6. 8: BP Player (Open Pocket Inventory Event)

From figure 6.8, are displaying how the user able to turn on and off the inventory.

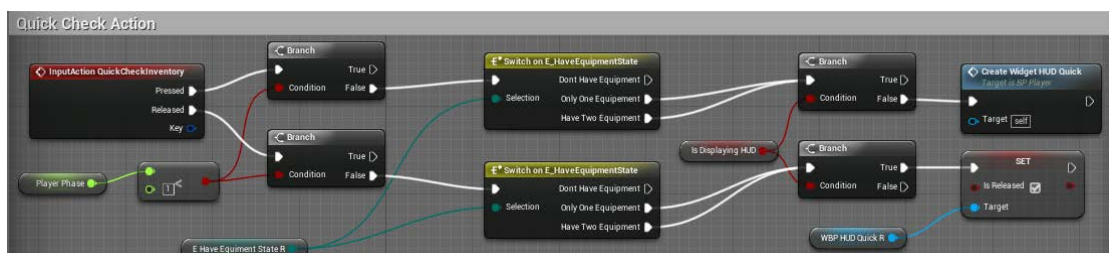


Figure 6. 9: BP Player (Quick Check Event)

Figure 6.9 is show how the user allows to quick check the items. The items include flash battery out.Andd syringe out.

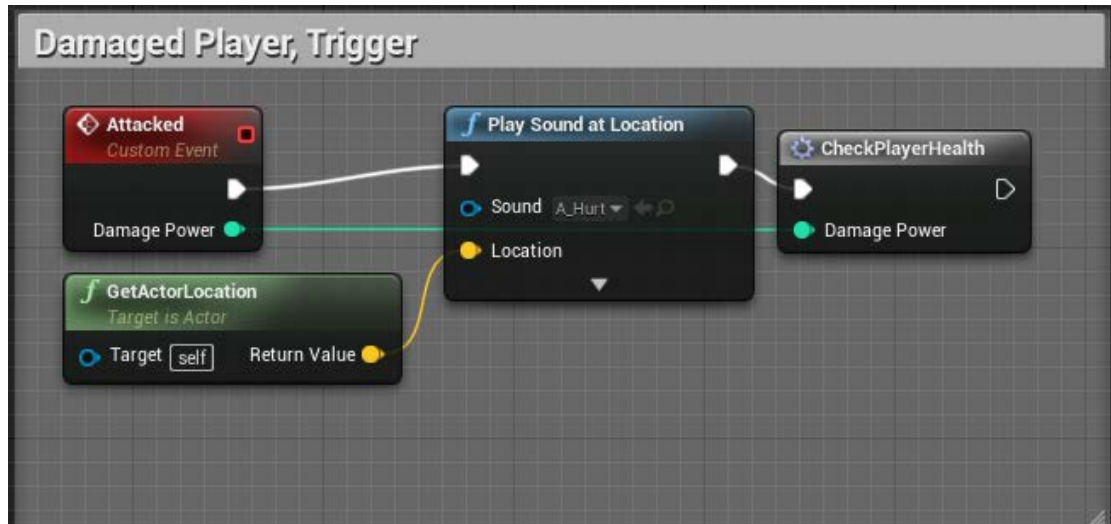


Figure 6. 10: BP Player (Damaged Trigger)

This image (6.10) represents when Enemy attack player. The sound effect will play and damage by unique attack damage power.

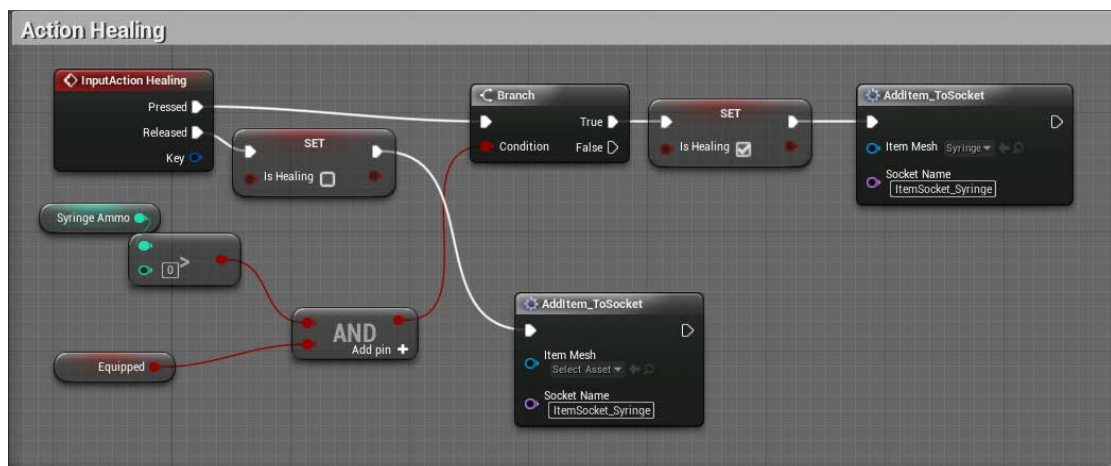


Figure 6. 11: BP Player (Healing Event)

Above Image, 6.11 represents how the user can Heal. First will identify the amount of the syringe is available, than continue to process the flow. This requires the user to hold the button or key rather than just press. Alternatively, else the whole action will be canceled and need to reset again.

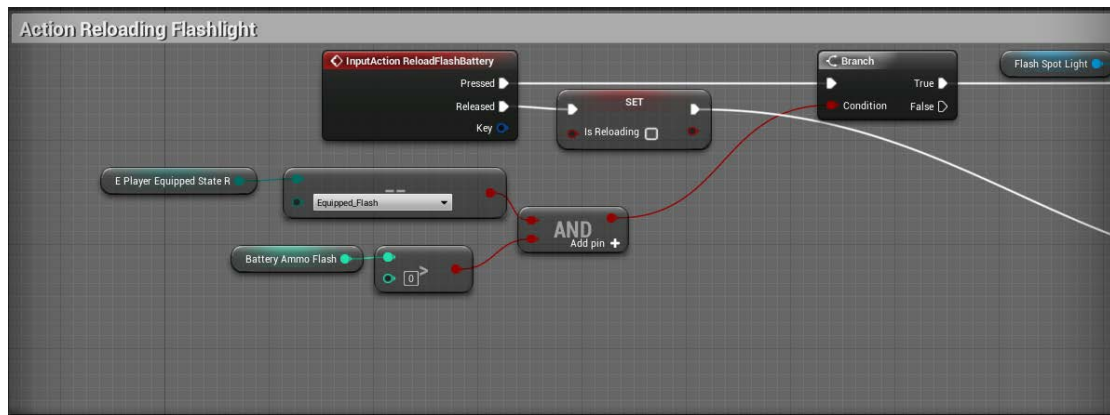


Figure 6.12: BP Player (Reload Event Part 1)

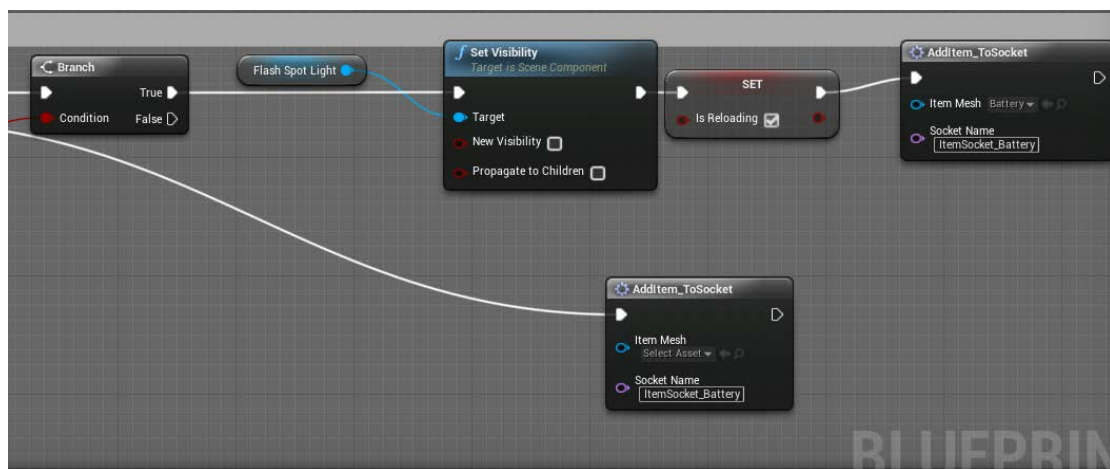


Figure 6.13: BP Player (Reload Event Part 2)

Based on Figure 6.12, and 6.13. Is display the code flow of reloading flash battery. First to identify the user are equipping flash, and the amount of the battery is available, then proceed the full of the process. This requires the user to hold the button or key rather than just press. Alternatively, else the whole action will be canceled and need to reset again.

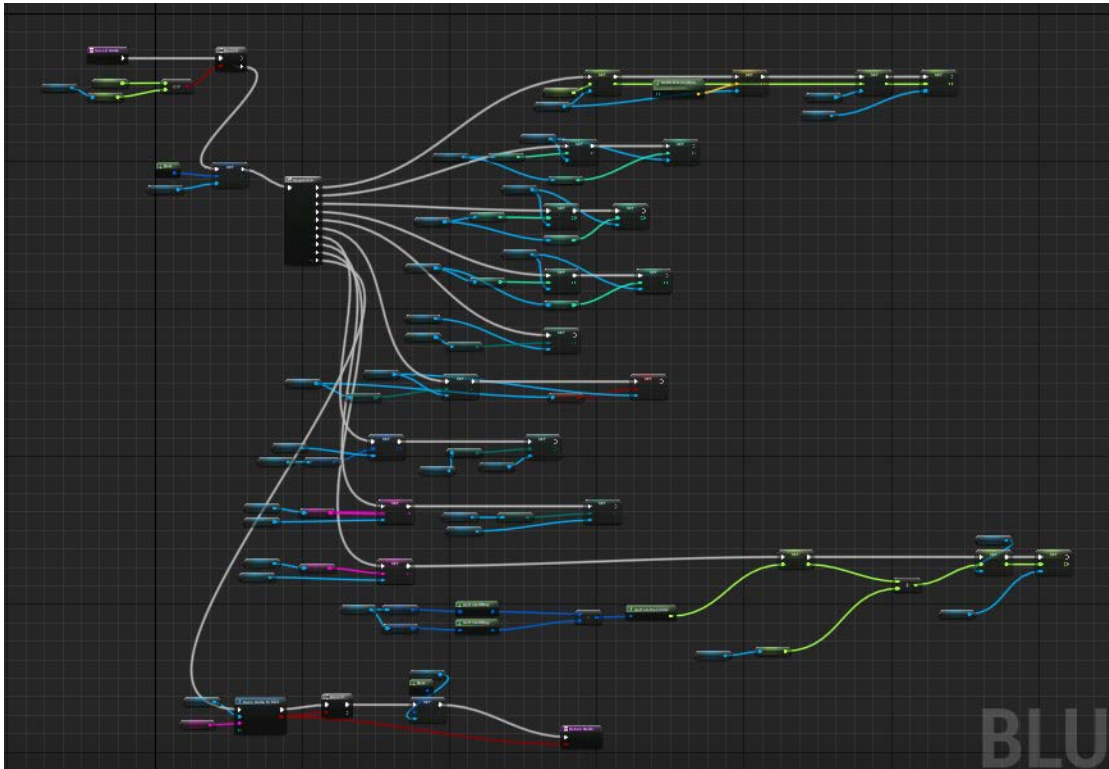


Figure 6.14: BP Checkpoint (Save Game Function)

From above the figure 6.14, is illustrate what of the data will be save into save the game object. More detail code in **Appendices B – Function Save Game**.

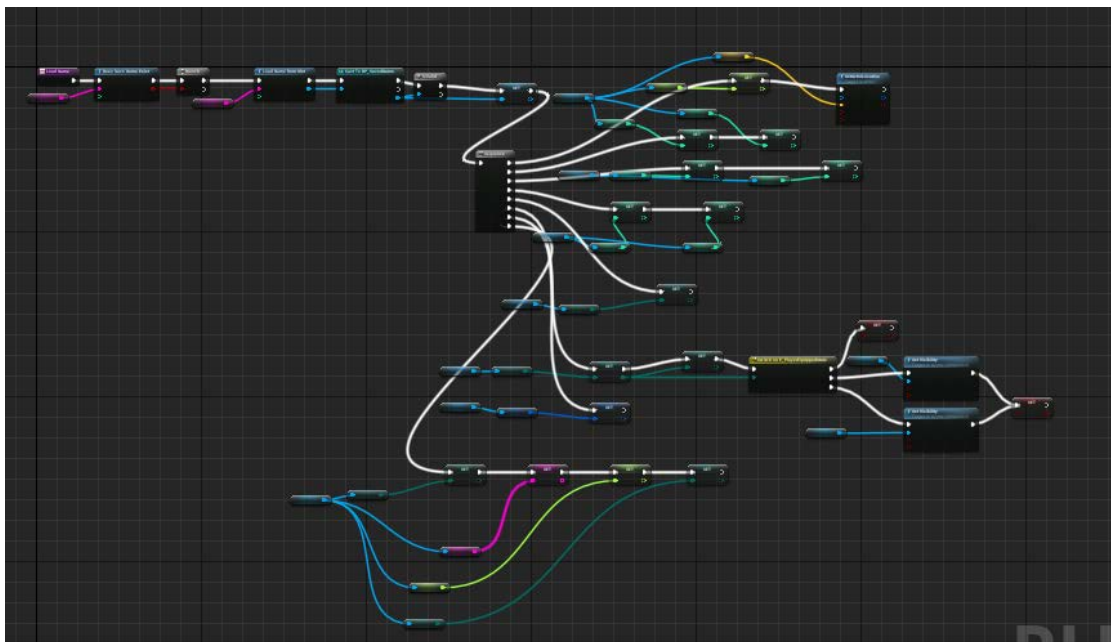


Figure 6.15: BP Player (Load Game Function)

This image (figure 6.15) shows how and what of the data will be retrieved from the game saved object, what of the object should take and loaded. Detail code on **Appendices B – Function Load Game**.

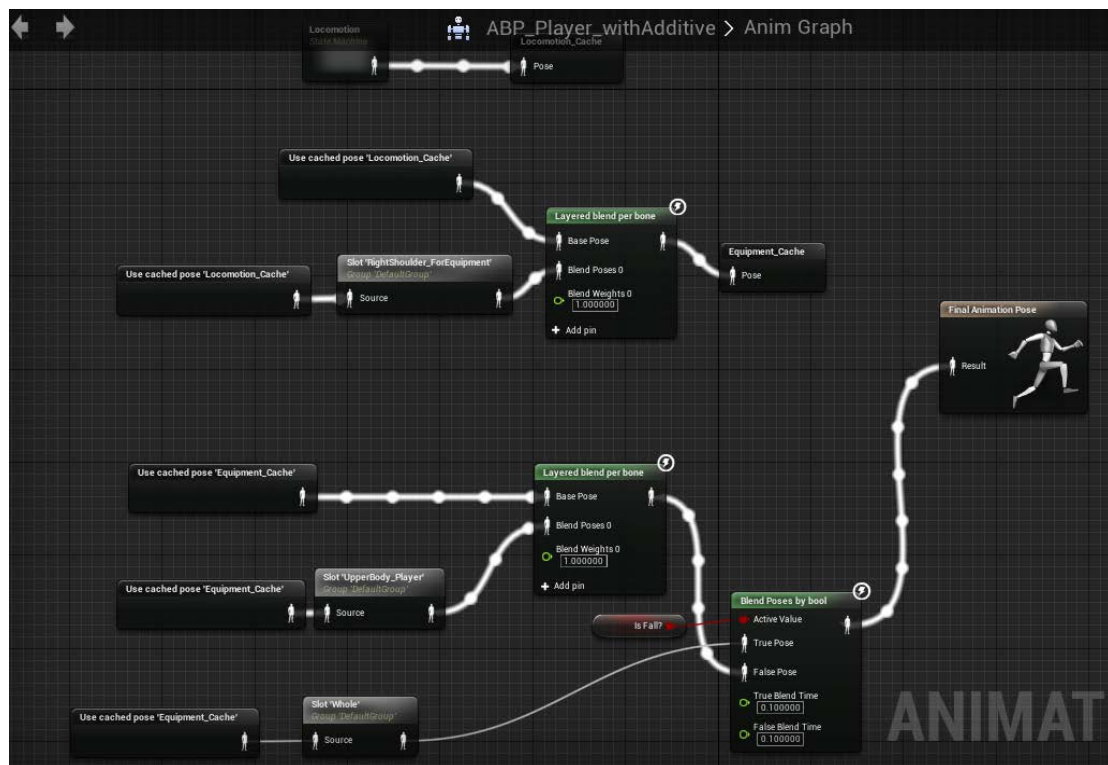


Figure 6. 16: ABP Player with Additive (Anim Graph)

Figure 6.16 is showing the overall animation how will process, such using two animations together however replace bone from one animation to other animation.

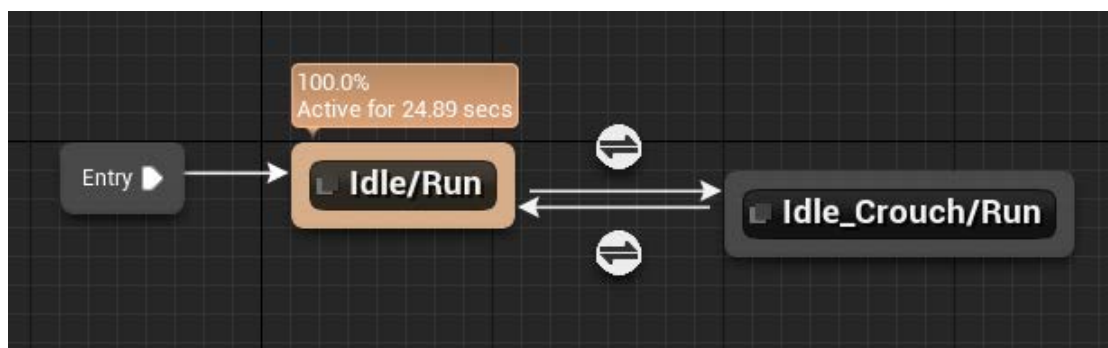


Figure 6. 17: ABP Player with Additive (Locomotion)

From image 6.17, Locomotion is generally how the animation flow, from normal (idle) to running, then continue to crouch and jump. Is a flow of the animation.

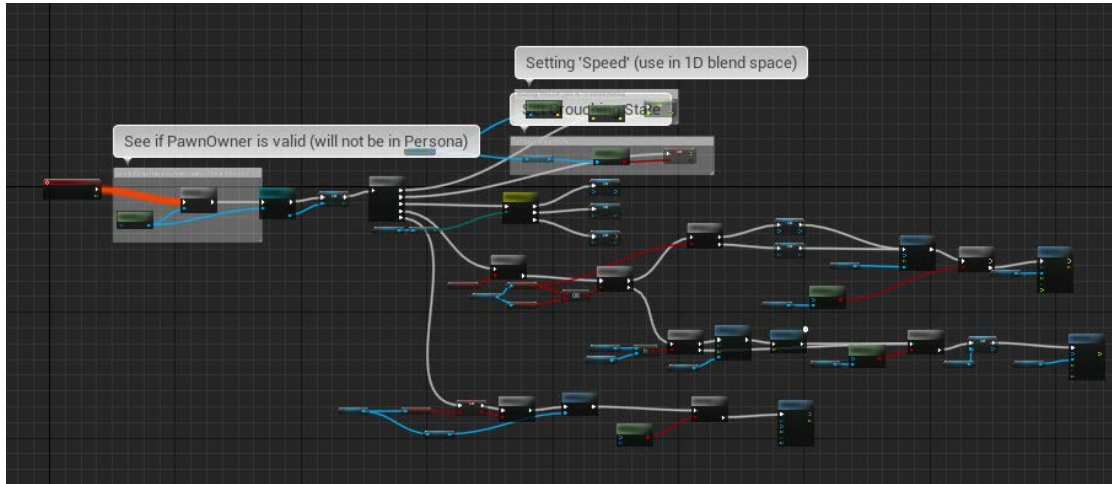


Figure 6. 18: ABP Player with Additive (Animation Event)

Image 6.18 is the overall of the Player animation event. For example, how the Player can receive data for Running, Crouching, Holding items, more detail will be available in **Appendices B – Animation Anim (Player)**.

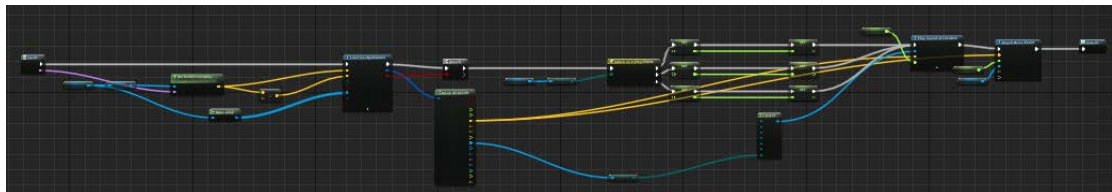


Figure 6. 19: ABP Player with Additive (Macro Footstep Song)

Figure 6.19 is a macro (private method or function), this macro is to play the sound effect of a footstep, based on Player Status. Either volume low or high. Detail code in **Appendices B – Footstep**.

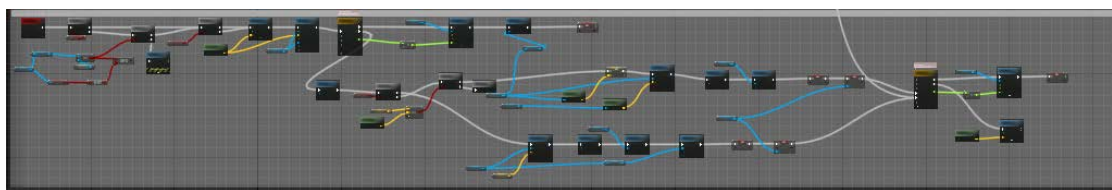


Figure 6. 20: BP Locker (Hide Event)

Image 6.20 is about how the player can perform the wedding event to hide into specifying furniture. Furniture such as wardrobe, and locker. More detail in **Appendices B – Hide Event**.

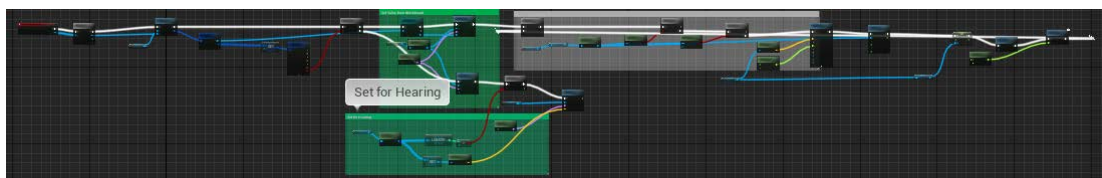


Figure 6.21: BP AI Controller School Girl (Perception Event)

This figure 6.21 is one of the enemy class, school girl. How her to get each perception like seen, hearing to get the data where the player is and continue acting. More detail on **Appendices B – Perception Event**.

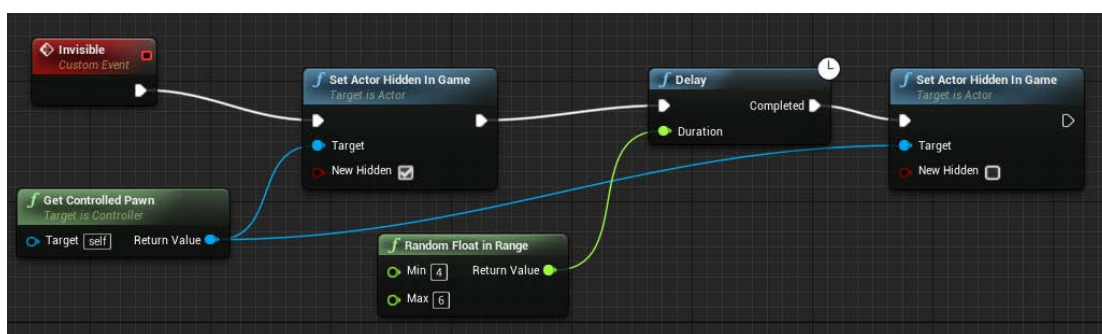


Figure 6.22: BP AI Controller School Girl (Invisible Event)

Figure 6.22, an event of this will activate when Enemy saw the player. Enemy self will invisible few second and restore back. To make the game perform in overall to be scarier.

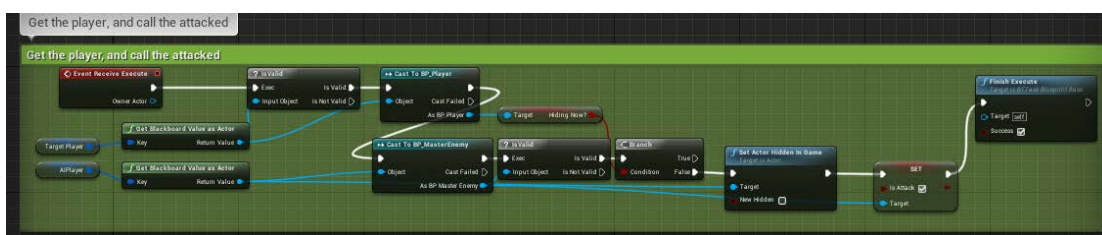


Figure 6.23: BTTask AttackPlayer

This figure 6.23 illustrate the Behaviour Task for Attack Player when Enemy is close enough to Player location, will be reset the invisible state to visible and enable the attack event to allow the following of code from Animation Blueprint for checking current can Attack the user and Play the Attack Animation.

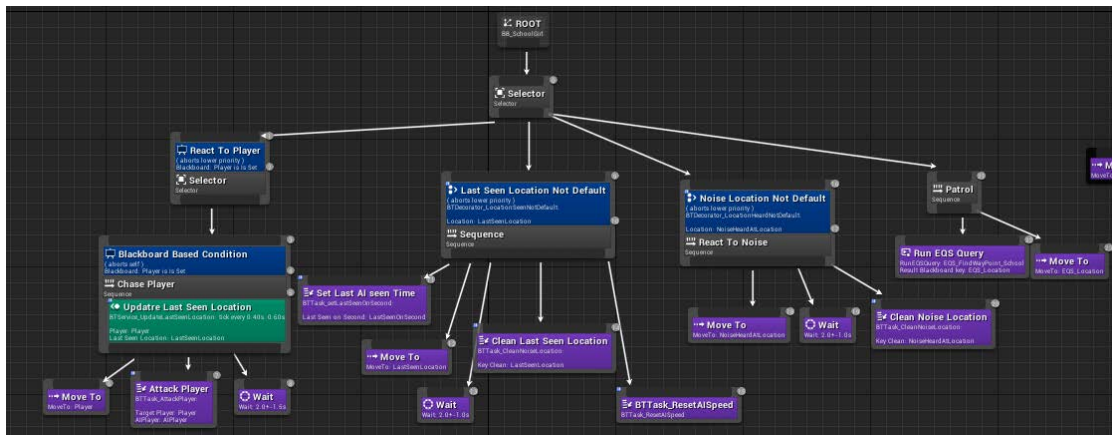


Figure 6. 24: BT School Girl

Figure 6.24, is the Behaviour tree of School Girl. If AI Controller is the sense, then Behaviour Tree will is a node to send all the information before Action. All action will perform here such start from chasing the player, then last seen location, move to noise location and continue patrol.

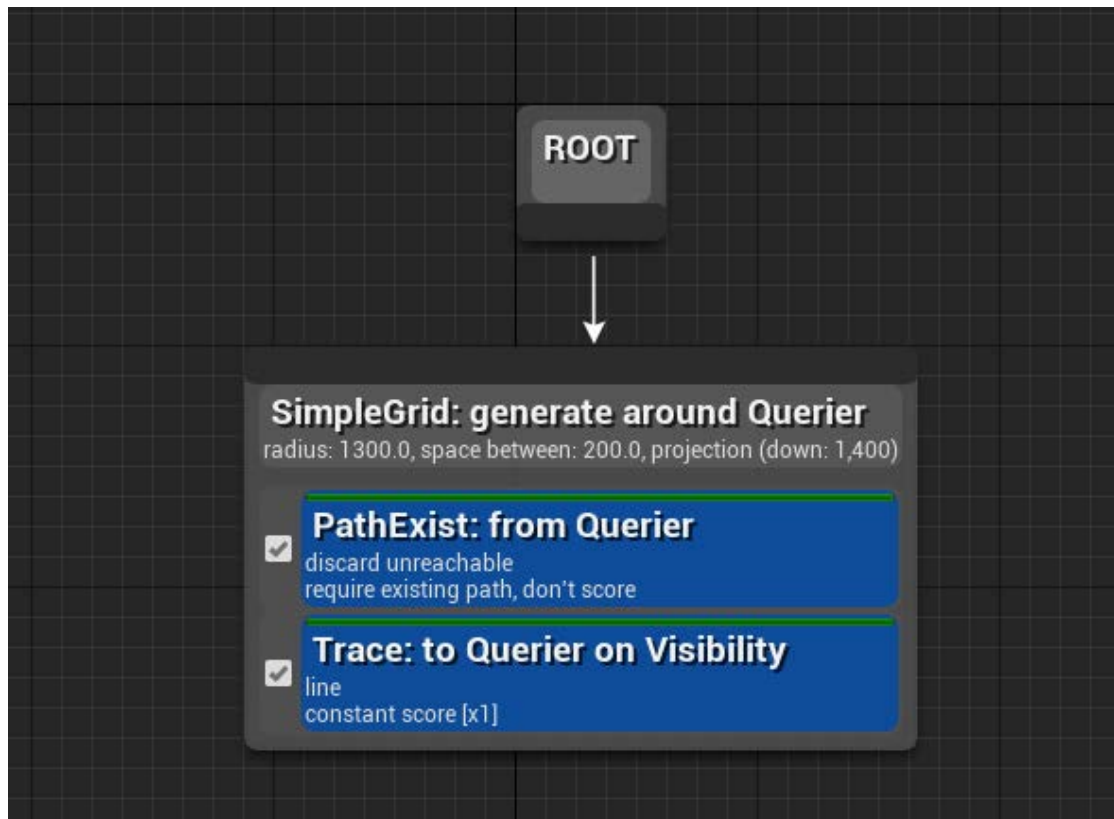


Figure 6. 25: EQS School Girl

Figure 6.25 is to find and trace the way, how the enemy can get the location for patrol.

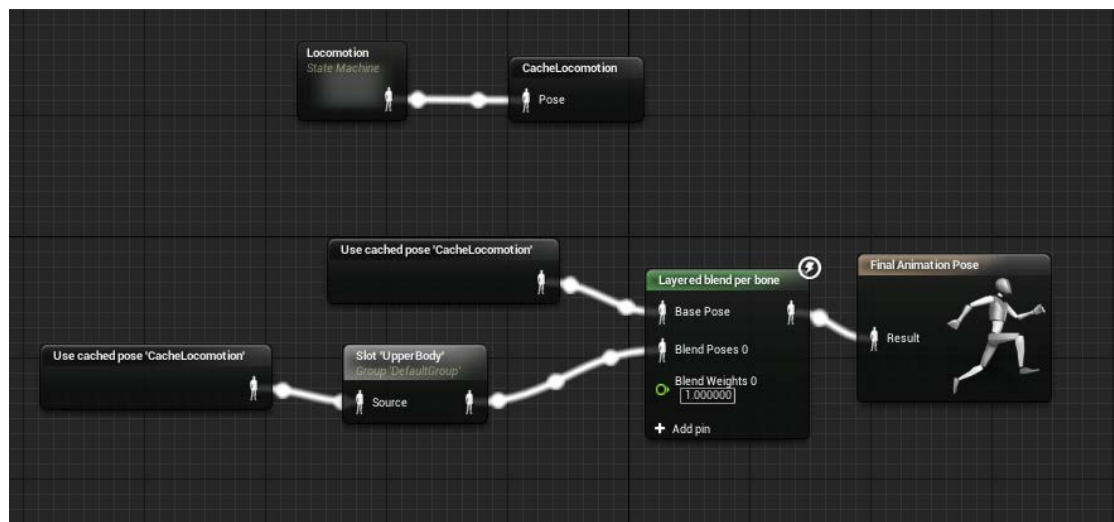


Figure 6. 26: ABP School Girl (Animation Graph)

Figure 6.26 is the Animation Blueprint of School Girl Animation Graph. Such when need to do the layer per bone (replace bone while two animation playing).

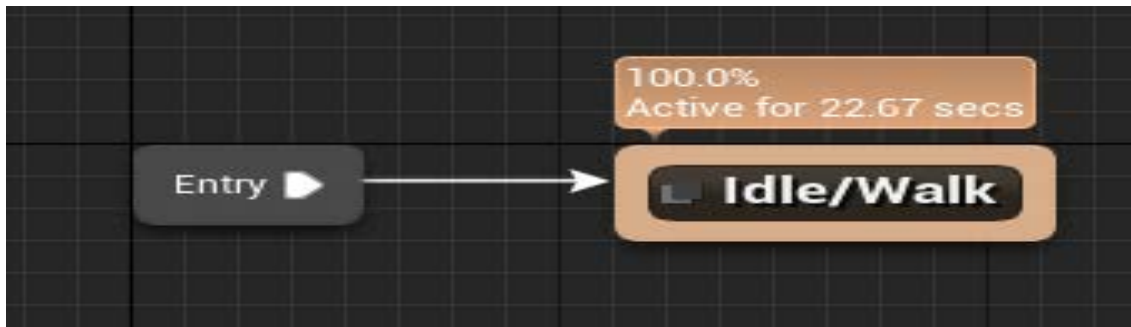


Figure 6.27: ABP School Girl (Locomotion)

This figure 6.27, is the Enemy Locomotion, is shortest than player since they need to stand and walk or run.

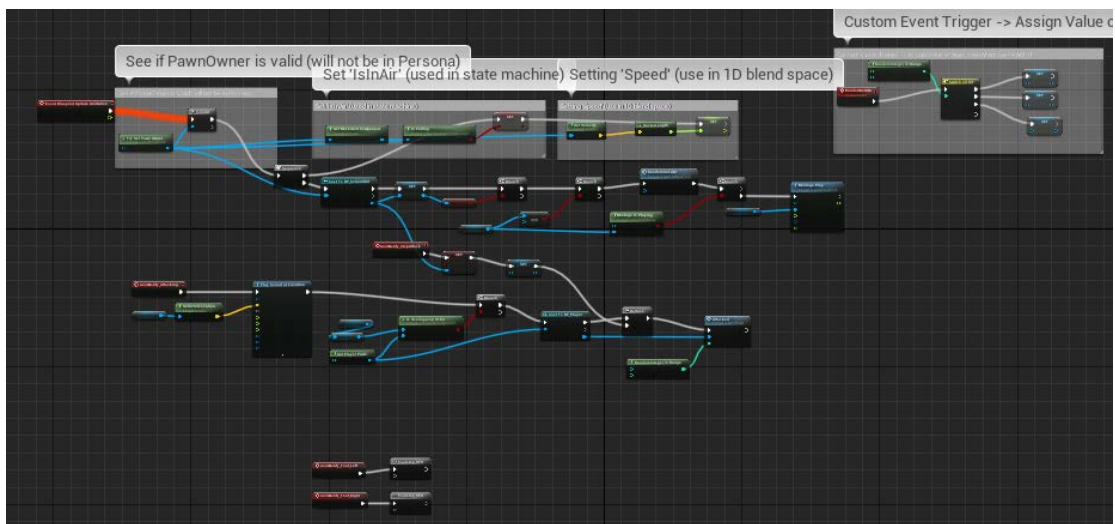


Figure 6.28: ABP School Girl (Animation Event)

From figure 6.28 is the School Girl Animation Blueprint, represent how to Enemy get the data for movement and also the play the attack animation. More detail on **Appendices B – Animation Anima (Enemy)**.

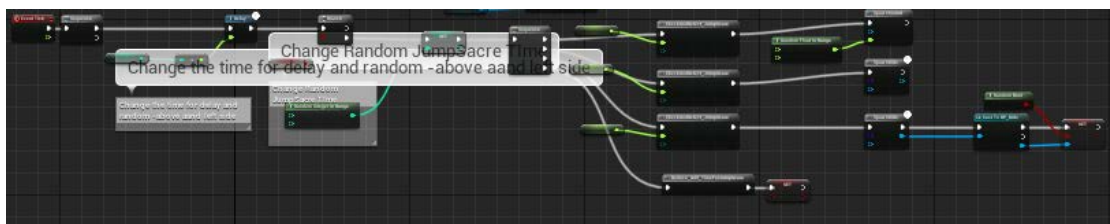


Figure 6.29: Level Blueprint – Night (Random Jump scare)

Figure 6.29 is Random Jump Scare will in between every 45 seconds to 60-second keep calling the macro, to identify which random jump scare will spawn and make fun to the player. The jumpscare included sound effect, spawn an enemy for exposed player location, an enemy to make more scary jumpscare effect. More detail in **Appendices B – Random Jump Scare**.

6.2.2 Game System Screenshot

6.2.2.1 Game



Figure 6. 30: Main Menu Screen

Figure 6.30 is our main menu screen, with animated background with natural sound effect. Is allow the user browse the official website to give feedback and also review the Credit scene to see who take the much effort in this game.



Figure 6. 31: Menu Option Screen

When User Press Start Game Button on Figure 6.30, Figure 6.31 will display, in this screen will allow the user to review he/she play time. The checkpoint and when last day play. It will be allowing every checkpoint to Start a new game again, however, Load Game only work at the Phase greater than 0.

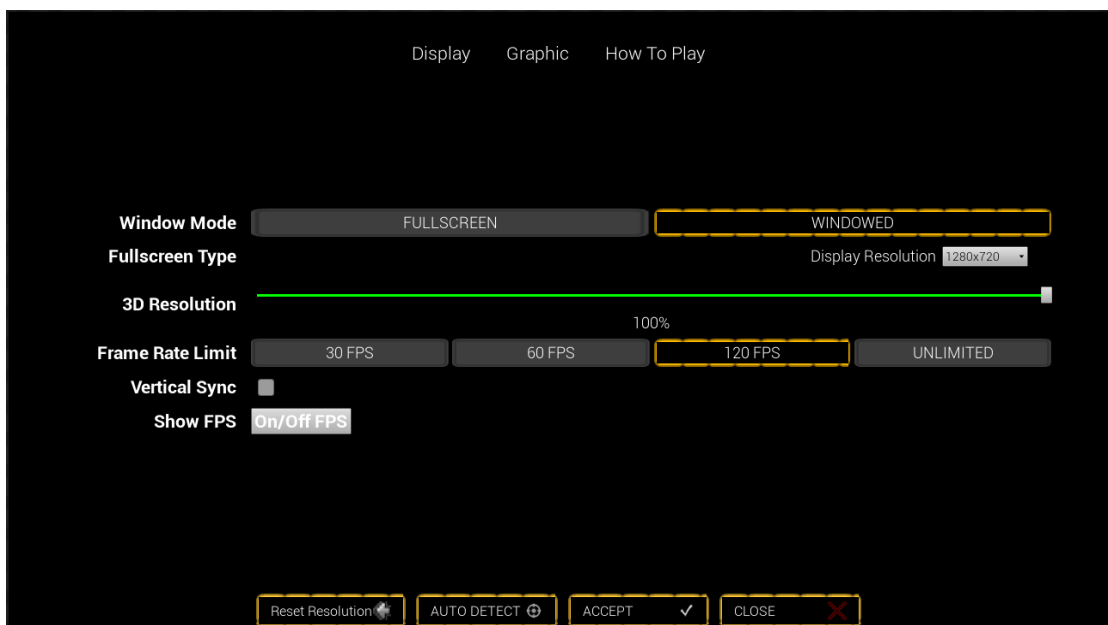


Figure 6. 32: Setting Option Screen (Display)

When the user clicks Setting, this options screen will display (Figure 6.32). Display Option allows the user to change the resolution, Window Mode and several features to

provide better performance in the game. Bottom buttons allow the user to accept the selection, close the screen option. Reset the default resolution, and Auto Detect the graphics option to get better performance.



Figure 6. 33: Setting Option Screen (Graphic)

Figure 6.33 is Graphic setting when User switch to Graphic option, it allows the user to modify the game graphic what want to present and it will affect in-game performance.

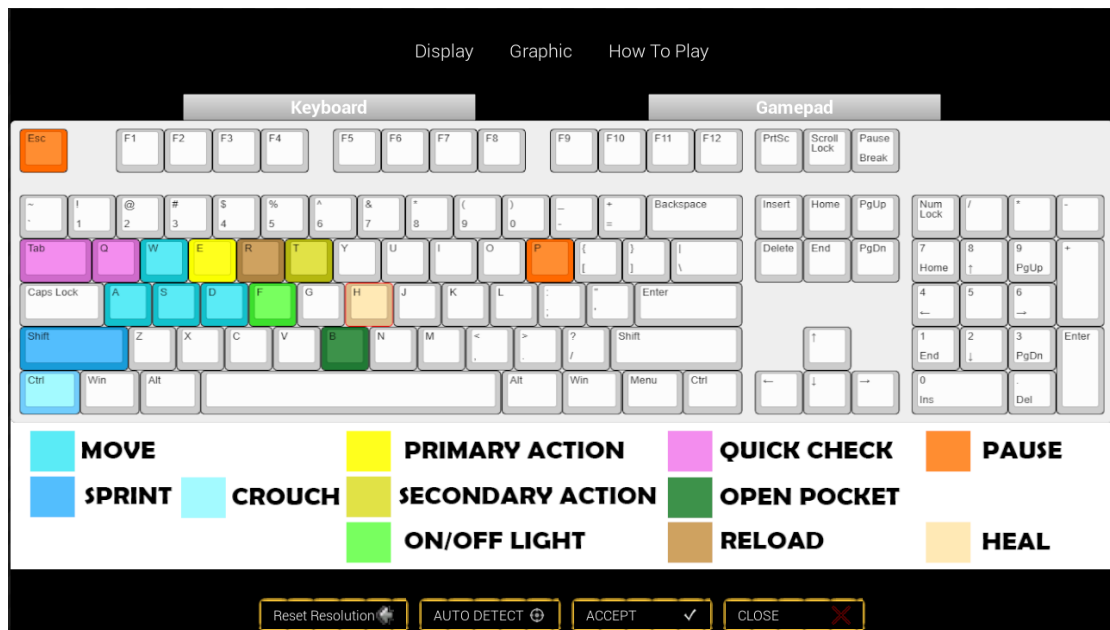


Figure 6. 34: Setting Option Screen (How To Play)

Figure 6.34 will give a guideline quickly to let people understand the control key and flow. This includes how the user uses the keyboard and mouse or controller to play the game.

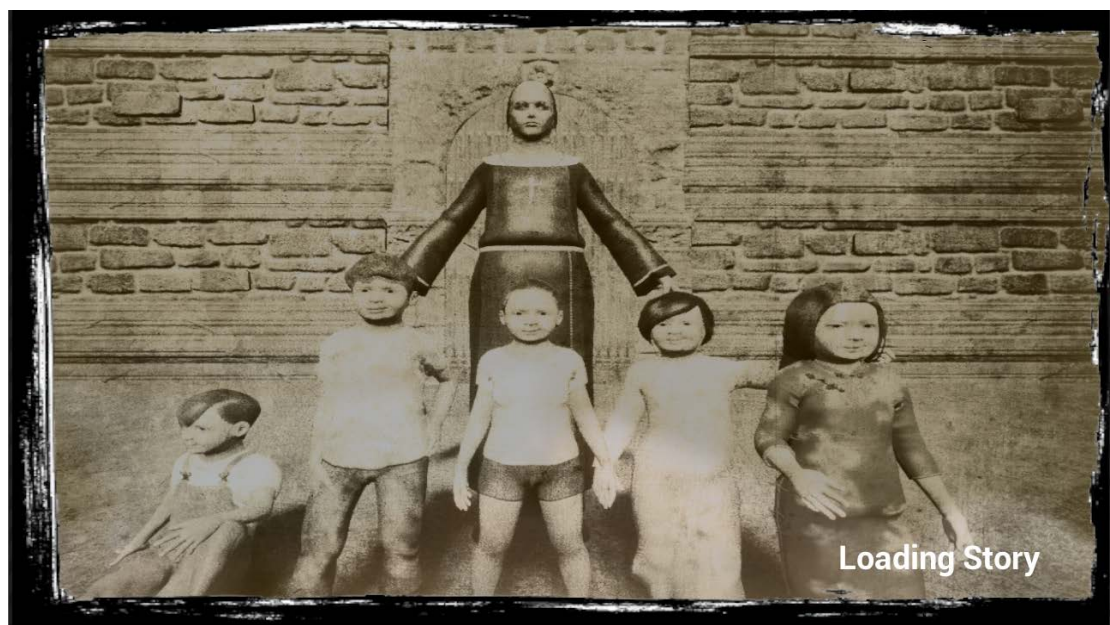


Figure 6. 35: Loading Screen

Figure 6.35 is a loading screen to give a message user take the time to wait and be patient.



Figure 6. 36: Pause Menu

Figure 6.36 is in-game Pause menu, allow user any time to pause the game, background will blur the environment. User allows to resume the game, reset the setting of option and return to the main menu screen.



Figure 6. 37: Key and Note

Figure 6.37 is two main item in the game; Key allow to unlock every locked door and note will give the hints, and tell you what of the happened in this village (Storyline).



Figure 6. 38: Pocket Key Inventory

This image (figure 6.38) show the user allow several keys to keep it however the pocket will full as well. Besides that, allow the user to use or drop the key.

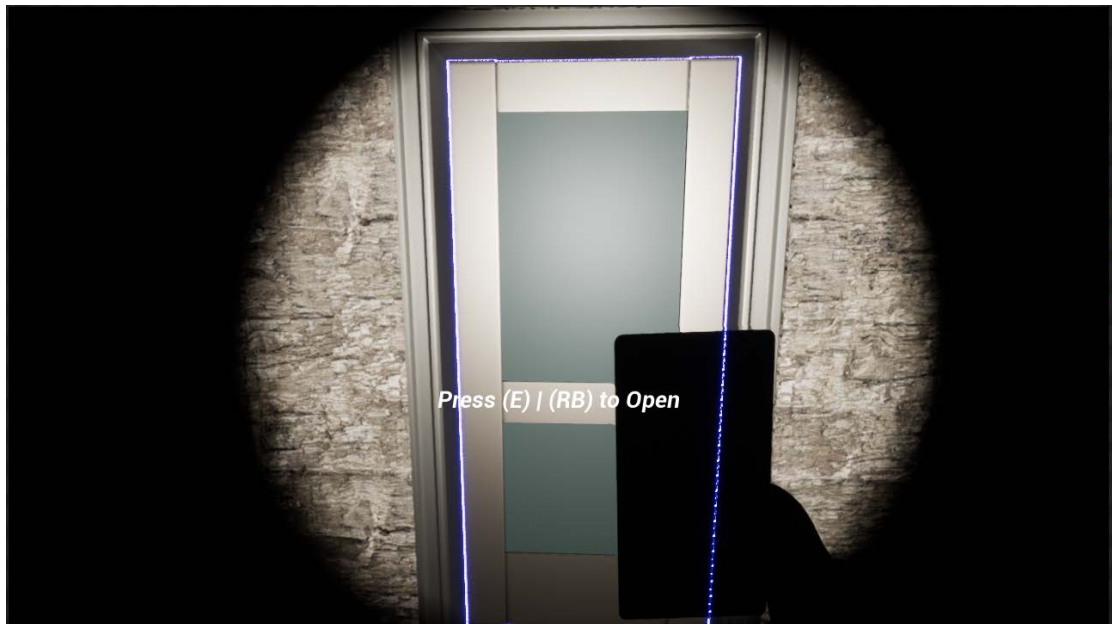


Figure 6. 39: Phone flashlight and Hints

Figure 6.39 represents the light source control by the user (On or Off), and the Hint will display when close to an interactable object or any emergency action such health is low.



Figure 6. 40: Flash light and Blood effect

Figure 6.40 image displays the better light source (Flash), and when hurt, the blood effect will surround the screen. Make it realistic



Figure 6. 41: Hide

Figure 6.41 is showing that Player is hiding from Wardrobe. A player able to hide in this game of 2 different objects, which Wardrobe and Locker.



Figure 6. 42: Game Over Screen

Figure 6.42 is Game Over screen, allow the user to Restart the last checkpoint or Back to Menu.

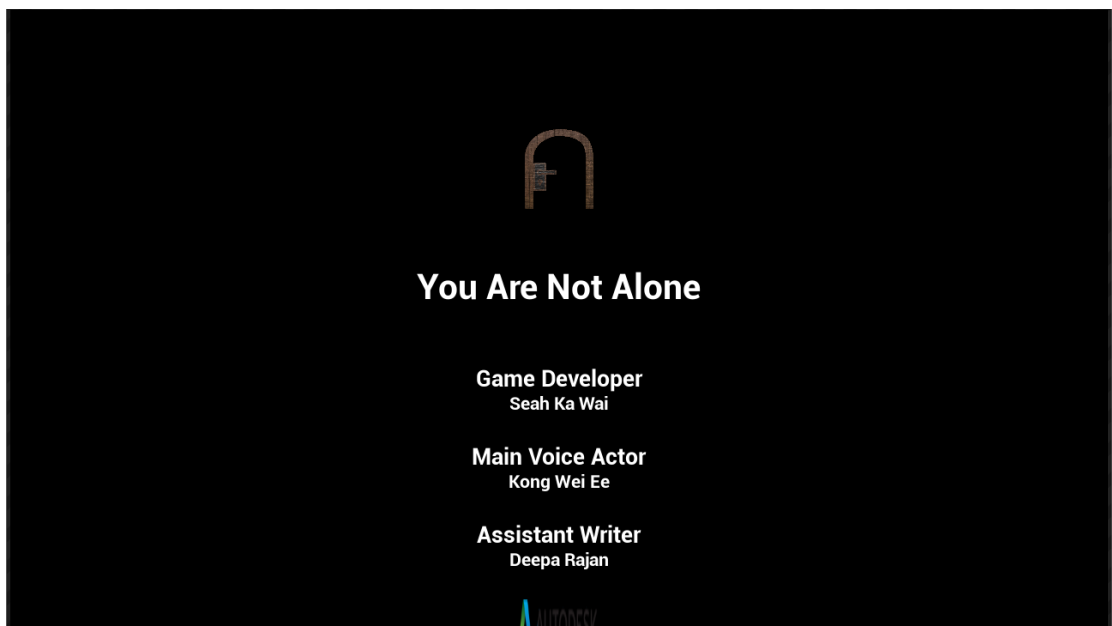


Figure 6. 43: Credit Scene

Figure 6.43 display the Credit Scene is present who and who's have spent the time in this game for help. The effort of the person involved in this system.

6.2.2.2 Building



Figure 6. 44: Motel in Daytime



Figure 6. 45: Slum in Daytime



Figure 6. 46: Police Station in Daytime



Figure 6. 47: School in Daytime



Figure 6. 48: Hospital in Daytime



Figure 6. 49: Church in Daytime

6.3 Conclusion

The implementation project is most important and most significant part of the whole project. Briefly explain the concept and the use of the code to quickly let the programmer understand the code flow and good on non-developer understand each action of concept. Developing application help to improve implementation skill and programming concept.

Sharing the several gameplay images to indicate how the game flow work and the building image will show what this environment in this game is. The gameplay image can indicate what of the object must in the game such Keys, and Notes. Key is to use the locked door to unlocked. The notes are allowing the user read and understand this game of the storyline. Each environment buildings are representing each Phase or Level in this game. Each environment also has the own obstacle and enemy want to face.

CHAPTER 7

TESTING AND IMPLEMENTATION

7.1 Introduction

This chapter will be discussing how the system be tested via the Testing Level. This testing level will have 4 step to test the game which Unit/Component testing, Integration testing, System testing and User Acceptance testing. List of testing will include Functional and Non-functional testing.

Functional Testing is the type of black-box testing based on specifications of software is tested. System tested by providing input, and results examined need to conform to functionality was intended.

Non-functional testing is involving requirements which nonfunctional but is essential such as user interface, and performance.

7.2 Testing level

7.2.1 Unit/Component Testing

Unit testing is verifying each part of software by isolating and perform the tests to demonstrate each component is correct regarding fulfilling the requirements and desired functionality.

This testing is performed at earliest stages of the development process, is executed by developers themselves before handing software over to the testing team.

The advantage is detecting errors in software early in the day by doing, to minimises software development risks, time and money wasted in having to go back and undone fundamental problems in the program once completed.

7.2.1.1 Menu Screen

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|---------------------------|--|--|---------------|---------|
| 1 | Play Button | Can launch to select save game | Switch to Game Saved option menu | As expected | No |
| 2 | Setting Button | Can launch option menu | Switch to the Setting option menu | As expected | No |
| 3 | Quit Button | Can Exit Game | Quit the application | As expected | No |
| 4 | Credit Button | Can play the credit scene | Credit Scene can be play | As expected | No |
| 5 | Web Button | Can open a web browser for Company website | Open browser to launch the Company URL | As expected | No |
| 6 | Display Button | Can open Display setting option | Switch to Display Setting option | As expected | No |
| 7 | Graphic Button | Can open the Graphics setting option | Switch to Graphic setting option | As expected | No |
| 8 | Guideline Button | Can open Guideline option | Switch to Guideline option | As expected | No |
| 9 | All Setting Option Button | Can Identify and Select the button | Display the effect when clicked and store the value | As expected | No |
| 10 | Accept Button | Can Pop up a window show which option selected | Display a confirmation widow allow the user to make a decision | As expected | No |
| 11 | Close/no Button | Can close the window or menu | Allow the user to close the option menu | As expected | No |
| 12 | Reset Resolution Button | Can reset the default resolution of user | Allow the user to reset the default resolution | As expected | No |

| | | | | | |
|----|------------------|---|--|-------------|----|
| 13 | New Game Button | Can start the game as the beginning phase | Allow user start the game from the first phase | As expected | No |
| 14 | Load Game Button | Can continue the last checkpoint stage | Allow the user to continue playing the last checkpoint phase | As expected | No |

Table 7. 1: Unit Testing (Main Menu)

7.2.1.2 Player

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|-------------------------|---|---|---------------|---------|
| 1 | Move | User can move | A player able to move | As expected | No |
| 2 | Running | The user can play in running mode | A player able to run | As expected | No |
| 3 | Crouch | The user can play in crouch mode | A player able to crouch | As expected | No |
| 4 | Toggle Light source | The user can turn on/off the light source | A player able to turn on/off the light source | As expected | No |
| 5 | Switch Equipment | The user can switch equipment | A player able to switch equipment | As expected | No |
| 6 | Pick up | The user can pick up items | A player able to pick up any items | As expected | No |
| 7 | Interacted Inspect view | The user can view the note | A player able to inspect view note | As expected | No |
| 8 | Reload Flash | The user can reload flash | A player able to reload the flash | As expected | No |
| 9 | Healing | The user can heal her self | A player able to heal | As expected | No |
| 10 | Quick Check | Slide in/out the inventory option | A player able to view the inventory | As expected | No |
| 11 | Pocket Inventory | Open pocket inventory | A player able to open pocket inventory | As expected | No |
| 12 | Close Inventory | Off inventory | A player able to close the pocket | As expected | No |

| | | | | | |
|----|----------------|--|--|-------------|----|
| 13 | Use Button | Use the key item | A player able to use the key | As expected | No |
| 14 | Drop Button | Drop the key | A player able to drop the key | As expected | No |
| 15 | Pause | Show Pause menu | Pause menu display and Pause the game | As expected | No |
| 16 | Resume Game | Return to Game | Continue to play | As expected | No |
| 17 | Setting Button | Open Setting option | Display setting option menu | As expected | No |
| 18 | Return Menu | Return to Menu | Return to main menu | As expected | No |
| 19 | Effect | Effect display such blood, tired | Display effect on controller camera | As expected | No |
| 20 | Damage | Damage user | The user will be damaged | As expected | No |
| 21 | Show hints | Show hints to the player | Display different hints to the user | As expected | No |
| 22 | Toggle Door | The user can open and close door | A player able to open and close any door | As expected | No |
| 23 | Hide | The user can hide in specify the place | A player able to hide into specifying the location | As expected | No |

Table 7. 2: Unit Testing (Player)

7.2.1.3 Enemy (AI)

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|-----------|--------------------------------|--|---------------|---------|
| 1 | Move | AI can move | AI able to move | As expected | No |
| 2 | Running | AI can run in a running mode | AI able to run | As expected | No |
| 3 | Attack | AI can attack the player | AI able attack player in range | As expected | No |
| 4 | Patrol | AI can patrol the random place | AI able to patrol the point within the range | As expected | No |

| | | | | | |
|----|--------------------|---|--|-------------|----|
| 5 | Perception (seen) | AI can detect player and chase | AI able to chase the player when seen | As expected | No |
| 6 | Perception (Hear) | AI can detect the noise location, and check | AI able to find the noise location, and check | As expected | No |
| 7 | Last seen location | AI able to move to last seen location when missing the player | AI able to search and go the last seen location | As expected | No |
| 8 | Toggle Door | AI can open and close door | AI able to open and close the door | As expected | No |
| 9 | Magic Trick | move hiding place fly on top | AI able to transform the hiding place on top of the user | As expected | No |
| 10 | Toggle Visible | AI can be invisible and visible | AI able to random invisible and visible to attack | As expected | No |

Table 7. 3: Unit Testing (Enemy)

7.2.1.4 Checkpoint

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|-----------|---------------------------|------------------------------------|---------------|---------|
| 1 | Save Game | Can save game | Able to identify and save the game | As expected | No |

Table 7. 4: Unit Testing (Checkpoint)

7.2.1.5 Scene

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|----------------------|------------------------------|--|---------------|---------|
| 1 | Perform Open Scene | Play Open scene | Play Open Video | As expected | No |
| 2 | Perform Ending Scene | Identify and Perform a scene | Able to identify which end should play | As expected | No |

Table 7. 5: Unit Testing (Game Scene)

7.2.2 Integration Testing

Integration testing is different test parts of the system in combination if work correctly together. By examining units in groups, any faults interact together can identify.

There are two ways to test about how different components of system function at their interface, bottom-up and top-down integration method.

Bottom-up integration testing builds on results of unit testing by testing higher-level combination units (called modules) in successively more complex scenarios.

Top-down integration testing, highest-level modules are tested first and progressively, lower-level modules are tested after.

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|------------------------|---------------------------------|--|---------------|---------|
| 1 | Main Menu (Start Game) | Game can start | Able to start the game (new/continue) | As expected | No |
| 2 | Main Menu (Setting) | Setting option able to change | Able to change all the option | As expected | No |
| 3 | Pause Menu | Pause the game | Able pause the game | As expected | No |
| 4 | Player (Movement) | Can move in a different mode | Able to move in a different mode with effect | As expected | No |
| 5 | Player (Action) | Can do all the action such take | Able to perform a Primary and Secondary action | As expected | No |
| 6 | AI (Enemy) | Can sense and do decision | Able to become advance AI | As expected | No |

| | | | | | |
|---|---------------|----------------------------|--|-------------|----|
| 7 | Saved/Load | Can save game, and load | Able to save game via checkpoint, load game via the menu | As expected | No |
| 8 | Perform Scene | Check and play scene movie | Identify and play ending movie | As expected | No |

Table 7. 6: Integration Testing

7.2.3 System Testing

System testing is all components of software tested as whole to ensure overall product meets the requirements.

System testing is significant step the software almost ready ship and can test in an environment which closes to the user will experience once deployed.

System testing enables testers to ensure product meets business requirements, as determine runs smoothly within the operating environment.

| Test Case ID | Test Case | Test to execute Test Case | Expected Result | Actual Result | Comment |
|--------------|------------------|----------------------------|----------------------------|---------------|---------|
| 1 | Application Open | Open Game | Able to run game | As expected | No |
| 2 | Test all again | Test all the feature again | Able work well in the game | As expected | No |

Table 7. 7: System Testing

7.2.4 User Acceptance Testing

User Acceptance testing is level in software testing process where product gave the green light. This testing aims to evaluate whether system complies with end-user requirements and if ready for deployment.

Testing team will utilise various methods, such test cases to test software and use results obtained which the system can improve.

The scope of acceptance testing ranges from finding spelling mistakes and cosmetic errors to uncovering bugs could cause significant failure in application.

7.2.4.1 Feedback from User via Questionnaire

This is the feedback from the user via the Questionnaire. To let the user test the game and fill the form, this questionnaire was uploaded on the Official Web. This action will bring the user easier to return the feedback about that game was the first impression and feeling.

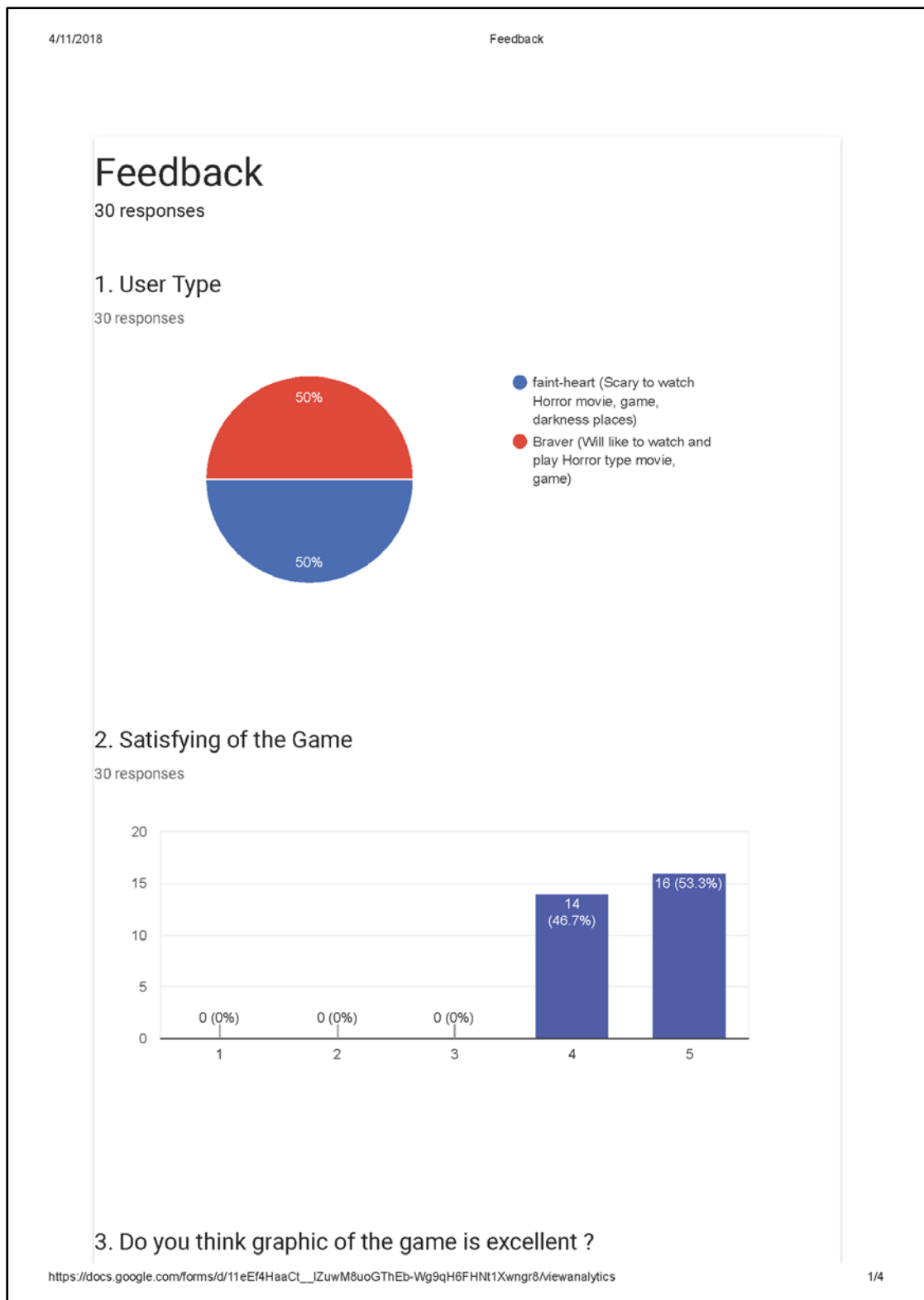


Figure 7. 1: Questionnaire Feedback (part 1)

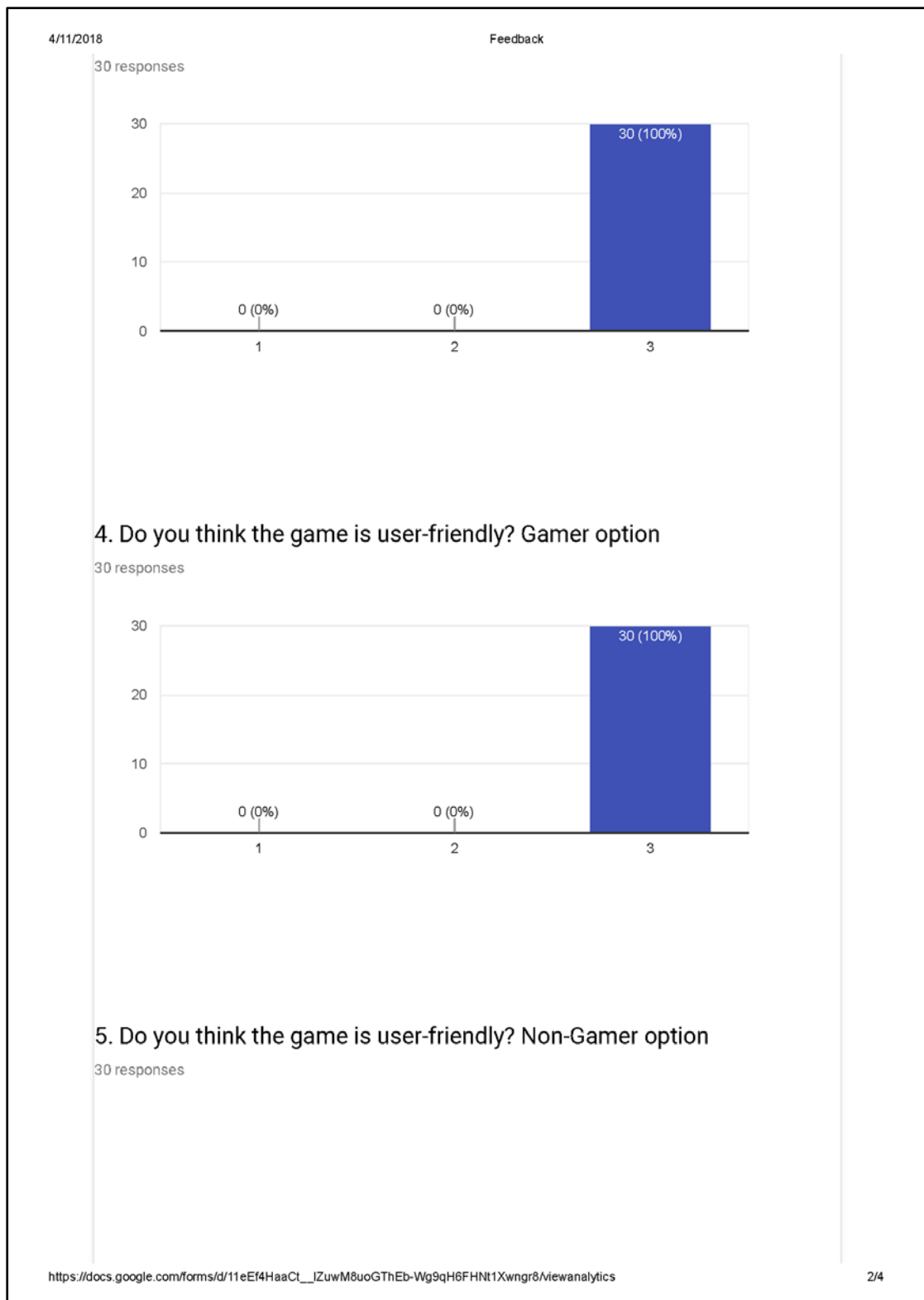


Figure 7.2: Questionnaire Feedback (part 2)

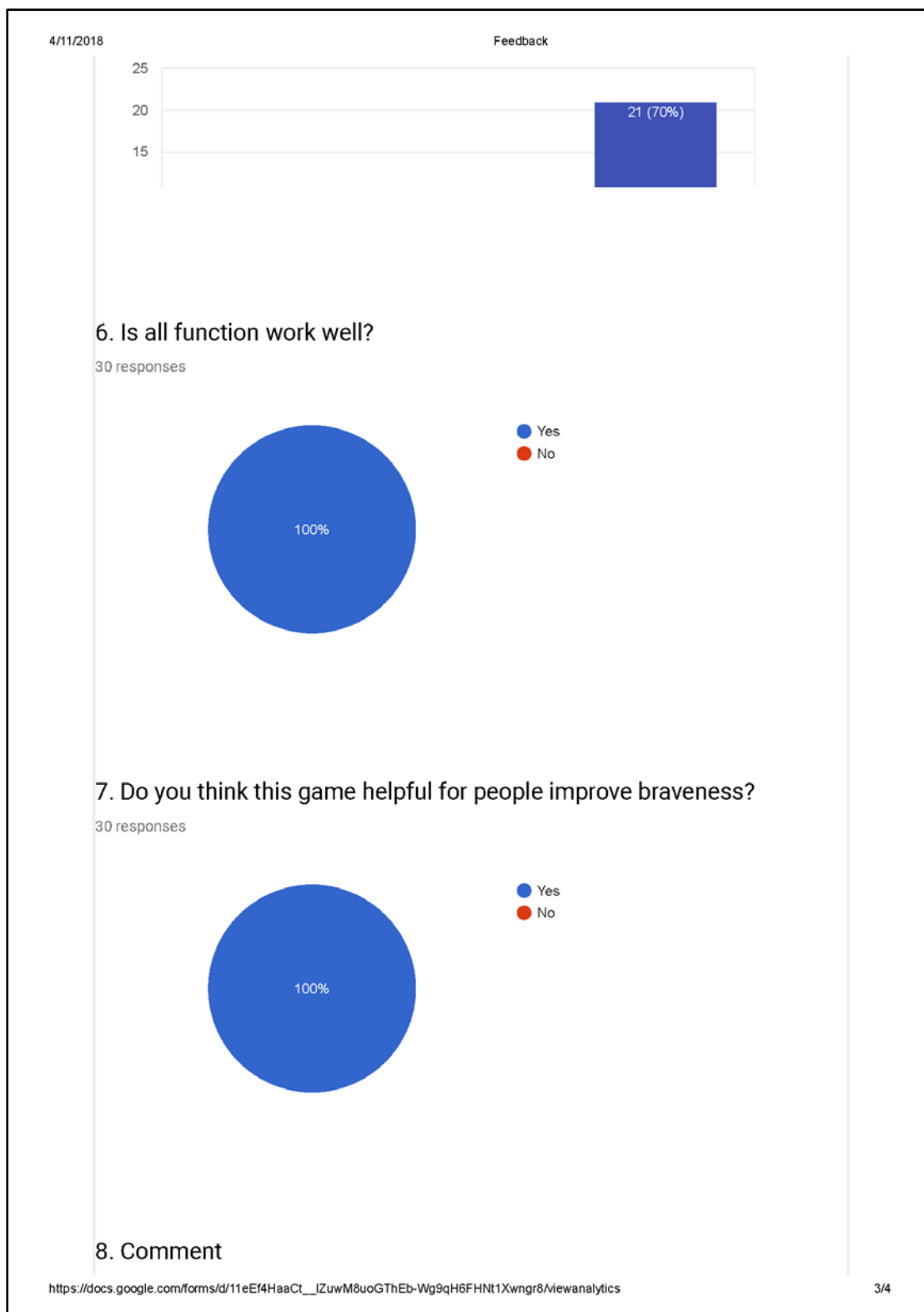


Figure 7.3: Questionnaire Feedback (part 3)

4/11/2018

Feedback

11 responses

No easy to play

Faint some time will occur cause more scary and afraid

Game quite difficult to win

Can really step out the fear after played

Nice jumpscare

Need time to learn

Game quite scary for braver as well

Scary

Fun to play, and enjoy the fear

Jump scare is nice

Quite Scare

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms

https://docs.google.com/forms/d/11eEf4HaaCt__IZuwM8uoGThEb-Wg9qH6FHnt1Xwngr8/viewanalytics

4/4

Figure 7.4: Questionnaire Feedback (part 4)

7.2.4.2 Feedback from User via Email and Message

This Feedback is collected from the Email Message and the Face to Face feedback. User allows to browse the Official Web to contact Developer what they comment, and

also the Game Website allows user quite to reply on the website without using any third party application to do that.

7.2.4.2.1 Require label hints (Solved)

For example, now need to get a key.

However, this game is more toward to real life, so that it requires the user to use the brain to think and play more than follow instruction and hints. Based on some user are like to follow hints, the system add-in the option allows the user to turn on/off the short hints.

7.2.4.2.2 Pause Menu will off the mouse cursor not allow the user to use mouse action for Pocket Inventory (Solved)

The user has tested a bug when Pocket Inventory is opening, and User pauses the game and returns to the game again. However, the mouse cursor disabled. Mean user is missing mouse cursor and cannot do any further action.

The problem had been solved, by added if the code to identify if the Pocket Inventory is displaying, the mouse cursor will not disappear once Pause Menu is off.

7.2.4.2.3 Require more Items (Solved)

The user feels difficulty play the game since the resources are limited.

To solved this problem, added more items in several areas so that user is a high possibility to get the resources (item) used.

7.3 Conclusion

Testing early and frequently is well worth effort. By adopting attitude constant alertness and scrutiny in projects, systematic approach to testing, the tester can pinpoint faults in system sooner, which translates less time and money wasted later.

Detecting software errors early is crucial since more effort needed to fix bugs when system near to launch, and due to interactive nature of components in the system, a small bug in particular part is hidden deep within layers of code can result in an effect that magnified several times over on system-level.

CHAPTER 8

CONCLUSION

8.1 Introduction

This chapter discusses the overall of the completed project of what I achieved in this project. The Outcome Summarization, the Strength and Weaknesses of Project, the Suggestion for Enhancement or Future Planning.

8.2 Outcome summarization

Based on this project, from the Chapter 1 is to do the planning of the Project and System What are the user want. When received the questionnaire feedback from the user. The result can be generating a clean image of what of the game are user want. On the Project, Background Section is more discussing what of the System (game) will be, what of the main feature and how the game process. Due to the observation and feedback can see how the target user will be facing a particular problem in their life. Moreover, also based on the comment given to current Movie and Game which have the same genre, what and how should improve. Those of the problem will be stated in the Problem Statement Section and how the solution should have solved these all problem will mention on Objective areas. The Scope of the project which How the system is playing the role, like present the story plot to the user. On the other hand, User Scope will be mention who is the Target User should be, and how the Audience allows to do in the system. Next, how the game will be attracted to the user, and how the game will bring the benefit to the user. Specify and briefly explain what of the System specification and software will be used for developing. Provided the Project Schedule to manage the own time management for return better result in the overall.

When the planning processed was done. When the user data have been collected. Next, is need to research about which existing system already released on the market. The reason to do this is that the Final System will improve the weakness or strength of the existing system and also come out specific unique feature of what of the current system does not have. Due to the reason can see what of the element must implement in the system and what should not. Additional, from the user comment or feedback from the existing system, can understand what is the objective of users want. Before the end of the research section, research about the Methodology to understand and implement for future implementation. The reason why to identify and select the most appropriate SDLC Methodology for own system is that we can follow the plan and reduce the time or resources had been wasted. Before concluding this Chapter 2, will be discussing more detail about each development tools of background.

In Chapter 3 will be discussing what of the Methodology had been selected and implement for future system implementation. Why in this project will be select the mythology to use? In each process explain detail how to relate to this project/system.

In Chapter 4, will be discussing several fact-finding methods, what are the advantages and disadvantage and which are the best choices to use in this project. Such method included Observation, Questionnaire and Interview. After that what of the System Requirements in this system. This has been split into two categories, Functional and Non Functional Requirement. Functional Requirement is mean that what of the user direct interaction in this system and what of the output will be present. Non Functional Requirement is what of the result will present by the game while the user is not doing anything. This more on how the game auto process the feature such saved game.

Chapter 5 is discussing the System Design, what are the modelling will be shown to the clients, users, or any people are interested in this project. The reason to do this is minimising the user spend of the time to read the whole report or system for understanding the whole complicated system. The user may not be the developer, so that is possible no idea any coding involved. Modelling diagram will present the brief of the process flow in the system and how executed. Besides that, several wireframes of the project will be provided; this wireframe is the idea of the system how should look like in the first phase.

Next, Chapter 6 Implementation and Programming System will overall be discussing what of the critical code to develop the feature on this system. Commonly several screenshots to present the code flow and short brief what the users of this code. Moreover, the System screenshot will be provided, to explain what of the critical thing commonly user will have interacted it and what is the mean on the screenshot.

Chapter 7 is discussing the Testing and Implementation, based on the Testing Level, started from Unit Testing until User Acceptance Testing. Unit Testing is tested each of the critical feature, such as in Main Menu Screen, the Button of Start Game. In Game Scene, the movement of Player. Next, the integration Testing is the combined all the feature into the specified class or object. Such all the feature of Player will be integrated into One Player Class. To do Unit Testing before Integration Testing is prevented any error occur which start integrate before a test or implement the function. System Testing will be the whole class, export or packaged into an executed file for User or Developer testing. Other meanings that are the integration become a Project or System. Moreover, Last, due to the exported file, is able the user to test the System, and based on the feedback from the user, we can identify and correct what should have needed to making a change.

The result of the project is satisfying for all the user. The is because the project has gained some reputation while doing the User Acceptance Test which user is given a different version of the game before deploying the complete project. The positive review and encouragement amongst user lead massive inspiration on developing the better game in future.

The game flow is better than expected, develop a game is different from the website, logical thinking and implementation will different to implement which require more information and test to apply to the game make sure not have any bug while or before playing.

After the testing phase will understand more detail of each stage, and based on the user review to make the game more improvement to fulfil user requirement.

8.3 Strength and weaknesses of project

The completion of the project shows the Strength and Weakness of project. This will outline and noted for future development or improvement.

8.3.1 Strengths

8.3.1.1 Immersive Graphics

In this Strength, Immersive Graphic is mean the game supports high texture of graphic, be able to lead the user to enjoy the high-end graphics game. Such as play game while feeling in real life. In this System, the Immersive Graphics will be the whole environment; this is because the game is 3D platform environment included much of foliage and lighting to get the better rendering result. Such reflection of Shadow in this game

8.3.1.2 Random Jump Scare

The game is providing random jump scare feature to improve the scariness since the jump scare is more will show on Movie style. A user playing the game can overcome the jump scare next time when watching the horror movie. Random Jump Scare in this project will be run in between every 45 seconds to 60 seconds or 1 minutes. Based on the random chance of percentage, if successful the percentage of the random jump scare will be a drop, however, the random jump scare cannot trigger on this time, the percentage to trigger will be increased. This project included 3 type of random jump scare, starting from scary beginner level to scary craziness level. First, the random sound, the first jump scare is random sound effect such laughing. This will interfere user concertation in the game and make them feel scared at that time. Second random jump scare has spawned a kid to surround the character (player), pointing out where you are, and make the noise to Enemy notice where are you hiding. Moreover, the third as final random jump scare is spawn a kid nearest you either player is looking at the kid or the kid saw you, the kid will either run forward to you or crawl to the player direction. Moreover, push the player fall, and some random jumpscare enemy will step in front of the player with around 200 milliseconds (0.2 seconds) and disappear it.

8.3.1.3 Advanced AI

The game has several AI (enemy) with unique play style, attack style. This action to make the game more dynamic and realistic. For example, a motel owner (boss) will have own attacking style without any equipment however a school girl will have a

baseball bat to attack the player. Each AI has own attack action, movement action or animation. Each AI also have specified damage power.

8.3.1.4 Multi Ending

This game is more toward in real life, and movie style. So that we include multi-ending scene allow user can play several time for understanding the detail of the story in this game. Multi Ending is based on how the user spends the time play the game, the ending start from bad, good, and secret or perfect ending. This reason why to do this is that if the user can collect all the note and understand the whole game story so will get the better result, however, if ignore all the valuable information and select to run and escape the place, will get the sad ending like the real life or movie style. Before figure out what of the happened and why will have this problem occur, but choose the priority on top are run to avoiding the bad happened, mostly the bad ending will like that.

8.3.1.5 Flexible to Check Update

The game can quickly tell the user when the game is updated by pressing the button on the Game Main Menu Screen (Check Update). If the game was updated, a small window will prop up and ask the user to update or not. If the user needs an updated version can click yes and will launch the browser to the official game web to download a new Updated Version of Game. This will make the user shortest they time since not every user will notify the news on the Social Media Network or another platform.

8.3.1.6 Free Download and Feedback anytime

Based on the game to allow every user can play, is created and published a website for the organisation and uploaded this game system. Next time, the user can anytime to download and install the game, enjoy and play. I have faced any problem or find some bug, can send the email via the website as well.

8.3.2 Weaknesses

8.3.2.1 Low-end PC cannot support

Since the game supports immersive graphics if the user is low-end devices are difficult to run smoothly in the game. Low specification of PC is difficult to play and enjoy this game; this is because the oldest hardware is not fully capable of running all the process at a faster speed. Due to the central reason in this game will have an immersive graphic, so that each frame will need the System to render for a better result.

8.3.2.2 A user with Color Vision Defect will difficult

Since the game is in darkness area, the environment is very dark. Who have colour vision defect possible can't thoroughly enjoy the game well. This is because of a survival horror game and this more toward on real life, an abandoned village. So that, the environment is messy and no any power supply. Is to illustrate the whole environment will be as dark as night like cannot see your hand of fingers.

8.4 Suggestion for enhancement/future planning

According to the weakness, and several ideas from user comment. This option will have been improving in future.

8.4.1 More Texture and Material for Low-end PC support

Due to the problem of the low-end computer cannot thoroughly enjoy the game, is required to create several materials, level of detail and texture to get more option for low-end devices user.

8.4.2 VR (Virtual Reality) experience

Due to the physiologist to help the patient who is scared in darkness area, provide and tell the patient keep wearing sunglass or any black glass at any time to get more comfortable next time. So that VR experience can bring the scariness of this game increased, and also can bring the user who is not brave enough can be a faster step out and overcome the problem.

8.4.3 DLC (Downloadable Content)

Since if the user feels the game of the story are well and enjoy. It is possible to add more new storyline about how the problem occurs when 18 years ago, and in-game how the male character to save the main character in this game.

8.5 Conclusion

In conclusion, the overall of this project need to start for analysing, planning, finding/researching, discussing or selecting best option, designing, implementing and testing. If we skip one phase will not do well on the system or delay the time for finishing it.

In overall, Chapter 1 is more on analysing and planning what want to do in this project. Chapter 2 is explaining more about what is finding or researching to improve in the project from existing system weakness. Chapter 3 is selected which methodology should implement in this whole project process; this can minimise the time spending for developing this project. Chapter 4 is to identify based on Chapter 1 and two research, which functional and nonfunctional interaction of user will have interacted. Next, Chapter 5 is the designing phase, to design the sketch how the game flow will be implemented and how the first look in the game system such Menu design or Building design (with 2D architecture design). Chapter 6 is more about how the system been implemented, such coding and the in-game interface. Chapter 7 will discuss how the testing phase should implement and what of the feedback user given. How to solved it. Moreover, the last chapter will discuss the overall what this Game Strength or Weakness is and What should enhance in future.

You Are Not Alone being absolute will be loved by teenage who like to challenge horror game, and also improve the brave. The game needs to keep improving from the user suggestion, and everything meet the user expectation and requirement. Even this game already accepted by most users. However, some user requires more future. Keep improving the game and add a feature, to avoid the game easily forget by the world. Future planning help to improve application better. It is crucial to make a summary of strength and weakness of project product so will become better product and user will continue use in future. The user can browse the Facebook page (<https://www.facebook.com/SKW.Studio.Page/>) to get the latest news in social media. Official Web (<http://skw-studio.com/>) will allow user understand what of the organisation are doing. Moreover, last, the Game website (<http://notalone.skw-studio.com/>) will allow the user to understand the subjective in this game, download, return feedback or comment and donation support.

REFERENCES

1. Adobe.com. (n.d.). *Buy Adobe Illustrator CC / Vector Graphic Design Software*. [online] Available at: <http://www.adobe.com/sea/products/illustrator.html> [Accessed 5 Oct. 2017].
2. Adobe.com. (n.d.). *Buy Adobe Photoshop CC / Best photo, image, and design editing software*. [online] Available at: <http://www.adobe.com/sea/products/photoshop.html> [Accessed 5 Oct. 2017].
3. Adobe.com. (n.d.). *Create 3D models, characters / Download Adobe Fuse CC (Beta)*. [online] Available at: <http://www.adobe.com/products/fuse.html> [Accessed 5 Oct. 2017].
4. Aguilera, M. d. & Mendiz, A., 2003. ideo games and education: (Education in the Face of a “Parallel School”). *Comput. Entertain*, pp. 1-10.
5. Audacityteam.org. (n.d.). *Features / Audacity®*. [online] Available at: <http://www.audacityteam.org/about/features/> [Accessed 5 Oct. 2017].
6. Bellotti, F., Berta, R., Gloria, A. D. & Primavera, L., 2009. Enhancing the educational value of video games. *Comput. Entertain*, pp. 1-18.
7. Best Selling Video Game. (2014). [image] Available at: <http://cdn2-www.playstationlifestyle.net/assets/uploads/2015/04/ESA-Genre-Sales-Data.jpg> [Accessed 5 Oct. 2017].
8. Bisson C, L. J., 1996. Fun in learning: the pedagogical role of fun in adventure education. *Journal of Experimental Education*, pp. 108-112.
9. Blake, J. & Goodman, J., 1999. Computer-based learning: games as an instructional strategy. *The Association of Black Nursing Faculty Journal*, pp. 43-46.
10. Chuang, T.-Y. & Wei-Fan, 2007. Effect of Computer-Based Video Games on Children: An Experimental Study. pp. 114-118.
11. Comment. (2013). [image] Available at: http://store.steampowered.com/app/252330/Slender_The_Arrival/ [Accessed 5 Oct. 2017].
12. Comment. (2013). [image] Available at: <http://store.steampowered.com/app/238320/Outlast/> [Accessed 5 Oct. 2017].
13. Comment. (2015). [image] Available at: http://store.steampowered.com/app/416590/Emily_Wants_To_Play/ [Accessed 5 Oct. 2017].
14. CSM (2016). *Software Development Life Cycle*. [online] Slideshare.net. Available at: <https://www.slideshare.net/manoharprasad/software-development-life-cycle-69516020> [Accessed 5 Oct. 2017].
15. Dorman, S. M., 1997. Video and computer games: effect on children and implications for health education. *Journal of School Health*, p. 133–138.
16. Emilywantstoplay.com. (2015). *Emily Wants to Play*. [online] Available at: <http://www.emilywantstoplay.com/> [Accessed 5 Oct. 2017].
17. En.wikipedia.org. (n.d.). *Survival horror*. [online] Available at: https://en.wikipedia.org/wiki/Survival_horror [Accessed 5 Sep. 2017].
18. Eriksson, U. (2014). *Differences between the different levels of tests*. [online] ReQtest. Available at: <https://reqtest.com/testing-blog/differences-between-the-different-levels-of-tests/> [Accessed 10 Feb. 2018].
19. Foundation, B. (n.d.). *About — blender.org*. [online] blender.org. Available at: <https://www.blender.org/about/> [Accessed 5 Oct. 2017].

20. GitHub Desktop. (n.d.). *GitHub Desktop*. [online] Available at: <https://desktop.github.com/> [Accessed 5 Oct. 2017].
21. GitHub. (n.d.). *kmkolasinski/AwesomeBump*. [online] Available at: <https://github.com/kmkolasinski/AwesomeBump> [Accessed 5 Oct. 2017].
22. Hebb, N. (2018). *What is a Flow Chart? | BreezeTree*. [online] Breetree.com. Available at: <http://www.breetree.com/articles/what-is-a-flow-chart.htm> [Accessed 15 Jan. 2018].
23. Istqbexamcertification.com. (2018). *What are Software Testing Levels?*. [online] Available at: <http://istqbexamcertification.com/what-are-software-testing-levels/> [Accessed 10 Feb. 2018].
24. Julie (2016). *Final documentation second year project*. [online] Es.slideshare.net. Available at: <https://es.slideshare.net/julieeustine1/final-documentation-second-year-project> [Accessed 5 Oct. 2017].
25. JV, D., BA, L., LL, H. & MS, C., 1996. *Instructional applications of computer games*. New York, ERIC Document Reproduction Service No. ED 394 500, pp. 8-12.
26. Lucidchart. (2018). *UML Activity Diagram Tutorial*. [online] Available at: <https://www.lucidchart.com/pages/uml-activity-diagram> [Accessed 20 Jan. 2018].
27. Lucidchart. (2018). *UML Use Case Diagram Tutorial*. [online] Available at: <https://www.lucidchart.com/pages/uml/use-case-diagram> [Accessed 15 Jan. 2018].
28. Lucidchart. (2018). *What is a Data Flow Diagram*. [online] Available at: <https://www.lucidchart.com/pages/data-flow-diagram> [Accessed 15 Jan. 2018].
29. Lucidchart. (2018). *What is a Flowchart*. [online] Available at: <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial> [Accessed 15 Jan. 2018].
30. Lucidchart. (2018). *What is Unified Modeling Language*. [online] Available at: <https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language> [Accessed 10 Jan. 2018].
31. Makehuman.org. (n.d.). *MakeHuman | Open source tool for making 3d characters*. [online] Available at: <http://www.makehuman.org/index.php> [Accessed 5 Oct. 2017].
32. Marchand, A. & Hennig-Thurau, T., 2013. Value Creation in the Video Game Industry: Industry Economics, Consumer. pp. 1-17.
33. Mark Hadley, B. (2013). *Slender: The Arrival - Now Available on Xbox One and Playstation 4*. [online] Slenderarrival.com. Available at: <http://slenderarrival.com/?page=betafaq> [Accessed 5 Oct. 2017].
34. McGonigal, J., 2011. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. s.l.: Penguin Books.
35. Midnight City, Majesco Entertainment (2013). [image] Available at: <http://cdn.akamai.steamstatic.com/steam/apps/252330/header.jpg?t=1482080188> [Accessed 5 Oct. 2017].

36. Mitchell, A., 2003. *Exploring the potential of a games-oriented implementation for m-portal*. London, s.n., pp. 19-20.
37. Most Popular 3D Modeling & Design Software for 3D Printing. (2017). [image] Available at: <https://i.materialise.com/blog/wp-content/uploads/2017/05/popular-3d-modeling-software-for-3d-printing.png> [Accessed 5 Oct. 2017].
38. Nahar, N. (2014). *Final project report of a game*. [online] Slideshare.net. Available at: <https://www.slideshare.net/NadiaIIT/final-project-report-of-a-game> [Accessed 10 Jan. 2018].
39. OpenLearn. (2018). *Models and modelling*. [online] Available at: <http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/models-and-modelling/content-section-2.1> [Accessed 10 Jan. 2018].
40. Prensky, M., 2001. *Digital game-based learning*. New York: McGraw-Hill.
41. Prensky, M., 2003. Digital game-based learning. *Computers in Entertainment*, p. 21–24.
42. Red Barrels (2013). [image] Available at: <http://cdn.akamai.steamstatic.com/steam/apps/238320/header.jpg?t=1447357854> [Accessed 5 Oct. 2017].
43. Red Barrels. (2013). *Outlast - Red Barrels*. [online] Available at: <https://redbarrelsgames.com/games/outlast/> [Accessed 5 Oct. 2017].
44. reddy, R. (2016). *uTest*. [online] Utest.com. Available at: <https://www.utest.com/articles/functional-testing-vs-non-functional-testing> [Accessed 10 Feb. 2018].
45. Relationship between GIT, B. (n.d.). *Relationship between GIT, Bit bucket and source tree*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/44150010/relationship-between-git-bit-bucket-and-source-tree> [Accessed 5 Oct. 2017].
46. Rouse, M. (2008). *What is flowchart? - Definition from WhatIs.com*. [online] WhatIs.com. Available at: <https://whatis.techtarget.com/definition/flowchart> [Accessed 15 Jan. 2018].
47. Rouse, M. (2015). *What is use case diagram (UML use case diagram)? - Definition from WhatIs.com*. [online] WhatIs.com. Available at: <https://whatis.techtarget.com/definition/use-case-diagram> [Accessed 20 Jan. 2018].
48. Ryte.com. (n.d.). *Waterfall Model*. [online] Available at: https://en.ryte.com/wiki/Waterfall_Model [Accessed 5 Nov. 2017].
49. SDLC. (n.d.). [image] Available at: https://www.tutorialspoint.com/sdlc/images/sdlc_stages.jpg [Accessed 5 Oct. 2017].
50. SKH Apps (2015). [image] Available at: <http://cdn.akamai.steamstatic.com/steam/apps/416590/header.jpg?t=1472246239> [Accessed 5 Oct. 2017].

51. Smartdraw.com. (2018). *Activity Diagram - Activity Diagram Symbols, Examples, and More*. [online] Available at: <https://www.smartdraw.com/activity-diagram/> [Accessed 20 Jan. 2018].
52. Smartdraw.com. (2018). *Flowchart - Everything You Need to Know About Flow Charts*. [online] Available at: <https://www.smartdraw.com/flowchart/> [Accessed 15 Jan. 2018].
53. Software Testing Fundamentals. (2018). *Software Testing Levels - Software Testing Fundamentals*. [online] Available at: <http://softwaretestingfundamentals.com/software-testing-levels/> [Accessed 10 Feb. 2018].
54. SourceTree. (n.d.). *SourceTree / Free Git GUI for Mac and Windows*. [online] Available at: <https://www.sourcetreeapp.com/> [Accessed 5 Oct. 2017].
55. Spiral. (n.d.). [image] Available at: https://www.tutorialspoint.com/sdlc/images/sdlc_spiral_model.jpg [Accessed 5 Oct. 2017].
56. Support. (2017). *Blender 2.79*. [online] Available at: <https://support.pdq.com/hc/en-us/articles/115001555592-Blender-2-79> [Accessed 5 Sep. 2017].
57. Test-institute.org. (2018). *Software Testing Levels - International Software Test Institute*. [online] Available at: https://www.test-institute.org/Software_Testing_Levels.php [Accessed 10 Feb. 2018].
58. Tom's Hardware. (2014). *New 'budget' build for 3D rendering and animations? - [Solved] - Systems*. [online] Available at: <http://www.tomshardware.com/answers/id-2044232/budget-build-rendering-animations.html> [Accessed 5 Sep. 2017].
59. Uml-diagrams.org. (2018). *Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors)*. [online] Available at: <https://www.uml-diagrams.org/use-case-diagrams.html> [Accessed 20 Jan. 2018].
60. Unrealengine.com. (n.d.). *Unreal Engine 4 is a professional suite of tools and technologies for building high-quality games and applications across a range of platforms*. [online] Available at: <https://www.unrealengine.com/en-US/features> [Accessed 5 Oct. 2017].
61. Visual-paradigm.com. (2018). *What is Activity Diagram?*. [online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> [Accessed 20 Jan. 2018].
62. Visual-paradigm.com. (2018). *What is Data Flow Diagram?*. [online] Available at: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/> [Accessed 15 Jan. 2018].
63. Visual-paradigm.com. (2018). *What is Use Case Diagram?*. [online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> [Accessed 15 Jan. 2018].

64. VModel. (n.d.). [image] Available at:
https://www.tutorialspoint.com/sdlc/images/sdlc_v_model.jpg [Accessed 5 Oct. 2017].
65. Waterfall. (n.d.). [image] Available at:
https://www.tutorialspoint.com/sdlc/images/sdlc_waterfall_model.jpg
 [Accessed 5 Oct. 2017].
66. Webster, L. (n.d.). *Visual Studio IDE*. [online] Visual Studio. Available at:
<https://www.visualstudio.com/vs/> [Accessed 5 Oct. 2017].
67. Wolf, M. J., 2001. *The Medium of the Video Game*. s.l.:University of Texas Press.
68. www.tutorialspoint.com. (2018). *Software Testing - Levels*. [online] Available at:
https://www.tutorialspoint.com/software_testing/software_testing_levels.htm
 [Accessed 10 Feb. 2018].
69. www.tutorialspoint.com. (2018). *UML Overview*. [online] Available at:
https://www.tutorialspoint.com/uml/uml_overview.htm [Accessed 10 Jan. 2018].
70. Xnagamedevelopment.blogspot.my. (2009). *Use Case Diagram*. [online] Available at:
<http://xnagamedevelopment.blogspot.my/2009/03/use-case-diagram.html> [Accessed 20 Jan. 2018].

APPENDICES

A

Sample Questionnaire (3D Game Development Part 1)

4/11/2018

3D Game Development Questionnaire

3D Game Development Questionnaire

This questionnaire is for a Final Year Project (FYP) on 3D Games Development, and it is entirely anonymous. By completing this survey, you have understood that your unknown results will be used to further this project and you agree that they may be used. Your opinion and feedback are critical to us. We will appreciate and many thanks.

* Required

1. Please Specify Your Gender ? *

Mark only one oval.

- ☐ Male
☐ Female

2. What age range do you fit into ? *

Mark only one oval.

- ☐ 0-15
☐ 16-21
☐ 22-35
☐ 35-50
☐ 50+

3. What is Your Occupation ? *

Mark only one oval.

- ☐ Student
☐ Self-employed
☐ Part-time employee
☐ Full-time employee
☐ Other: _____

4. Do you like to play game ? *

Mark only one oval.

- ☐ Yes
☐ No

5. Which(if any) of the following do you own and play games on: *

Check all that apply.

- ☐ PC
☐ Video Game Console (Such as PlayStation, Xbox, Nintendo Wii)
☐ Portable Video Game Console (Such as PS Vita, Nintendo Switch)
☐ Mobile Devices
☐ Other: _____

Sample Questionnaire (3D Game Development Part 2)

4/11/2018

3D Game Development Questionnaire

6. Based on the Question 6, which device are you more prefer to play game ? *

Mark only one oval.

- ☐ PC
- ☐ Video Game Console
- ☐ Portable Video Game Console
- ☐ Mobile Devices
- ☐ Other: _____

7. How much per game do you spend on average on computer games ? *

The currency in Dollar sign (\$)

Mark only one oval.

- ☐ \$0-\$10
- ☐ \$11-\$20
- ☐ \$21-\$30
- ☐ \$31-\$40
- ☐ \$41-\$50
- ☐ \$51+

8. Where do you purchase your games ? *

Can select more than one

Check all that apply.

- ☐ Official Website
- ☐ Digital Distribution Platform (Such as STEAM, Origin)
- ☐ Online Shops
- ☐ In Store
- ☐ Other: _____

9. Do you prefer to buy computer game on Digital Distribution Platform ? *

Mark only one oval.

- ☐ Yes
- ☐ No

Sample Questionnaire (3D Game Development Part 3)

4/11/2018

3D Game Development Questionnaire

10. Which is your favorite type of Game ? *

Mark only one oval.

- ☐ Arcade
☐ Action
☐ Driving/Racing
☐ First Person Shooter
☐ Horror
☐ MMORPGS
☐ Platform
☐ Puzzle
☐ Real-Time Strategy
☐ Retro
☐ Role-Playing
☐ Simulation
☐ Survival
☐ Other: _____

11. What compels you most to buy a computer game? *

Can select more than one

Check all that apply.

- ☐ Attractive Packaging
☐ Cheap
☐ Friend Recommendations
☐ Good Gameplay
☐ Good Graphics
☐ Good Reviews
☐ Innovative Control Systems
☐ Multiplayer Features
☐ New Release
☐ Published by a Reputable Company
☐ Sequel to another game
☐ Other: _____

12. Do you interested in Survival Horror Game ? (Like, Outlast) *

Mark only one oval.

- ☐ Yes
☐ No

Sample Questionnaire (3D Game Development Part 4)

4/11/2018

3D Game Development Questionnaire

13. Which features do you prefer for Survival Horror Game ? *

Can select more than one

Check all that apply.

- ☐ First-person perspective
- ☐ Third-person perspective
- ☐ Player allow to attack
- ☐ Limited Resources (Such as Battery of Flash Light)
- ☐ Random JumpScare
- ☐ Other: _____

14. What feature do you think should add to Survival Horror Game ? (Such as Virtual Reality [VR] mode) *

15. In your own words, what would you like to see most in a computer game? Or What would your ideal game be?

Optional

Last Question, the answer can allow us to track your record, will enable us to send an updating news to you, and you will be chosen as one of the lucky fellows to receive the game when had been developed and published

16. Email Address (Optional)

Powered by
 Google Forms

Sample Questionnaire (User Feedback Part 1)

4/11/2018

Feedback

Feedback

This is the feedback form for the User play and test the game You Are Not Alone

* Required

1. User Type *

Determine which user group who you are? Like the genre of Horror type or Not ?
Mark only one oval.

- ☐ faint-heart (Scary to watch Horror movie, game, darkness places)
- ☐ Braver (Will like to watch and play Horror type movie, game)

2. Satisfying of the Game *

How satisfy you play and test the game ?
Mark only one oval.

| | | | | | | |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
| | 1 | 2 | 3 | 4 | 5 | |
| No Satisfied | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | VerySatisfied |

3. Do you think graphic of the game is excellent? *

The graphic of this game would you like ?
Mark only one oval.

| | | | | |
|----|-----------------------|-----------------------|-----------------------|-----|
| | 1 | 2 | 3 | |
| No | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Yes |

4. Do you think the game is user-friendly? Gamer option *

For Gamer, easier to control ?
Mark only one oval.

| | | | | |
|----|-----------------------|-----------------------|-----------------------|-----|
| | 1 | 2 | 3 | |
| No | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Yes |

5. Do you think the game is user-friendly? Non-Gamer option *

For Non Gamer, easier to control ?
Mark only one oval.

| | | | | |
|----|-----------------------|-----------------------|-----------------------|-----|
| | 1 | 2 | 3 | |
| NO | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Yes |

6. Is all function work well? *

Is all the button, function work well ??
Mark only one oval.

- ☐ Yes
- ☐ No

Sample Questionnaire (User Feedback Part 2)

4/11/2018

Feedback

7. Do you think this game helpful for people improve braveness? *

As user, would you think this game can help those people improve braveness ?
Mark only one oval.

☐ Yes☐ No

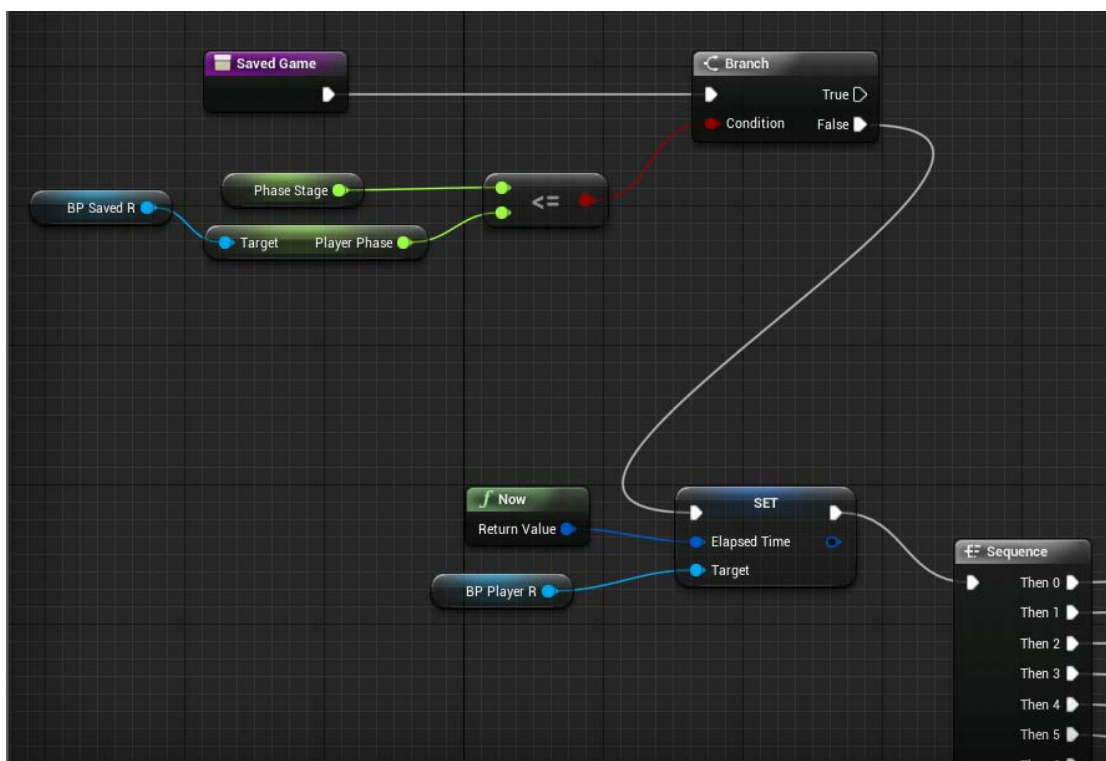
8. Comment

Comment or Feedback about the Game play

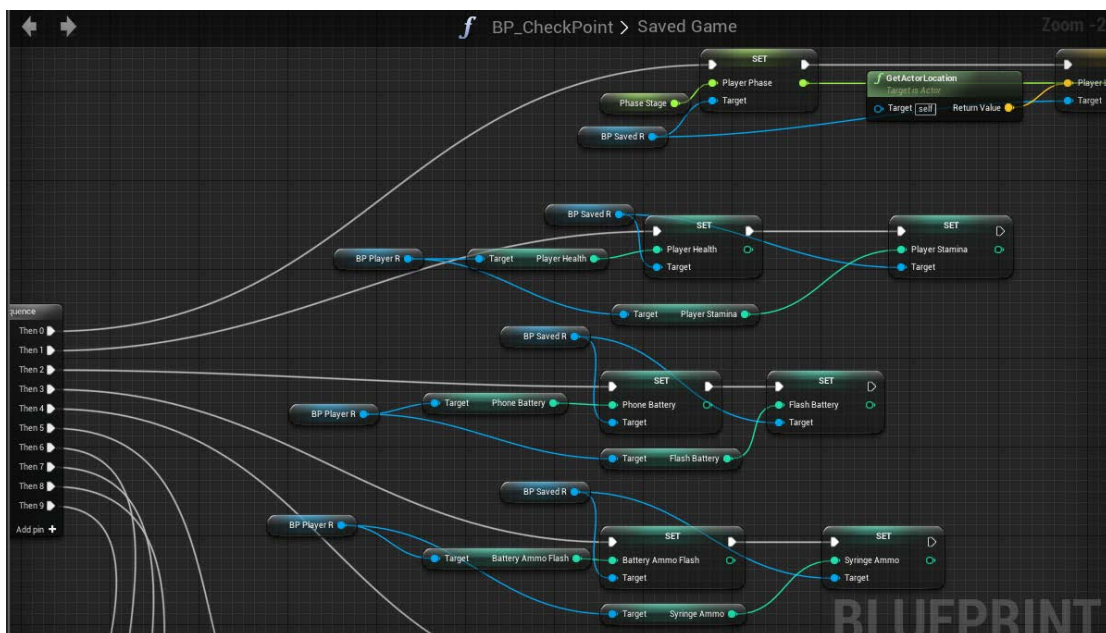
Powered by
 Google Forms

B

Function Save Game (Part 1)



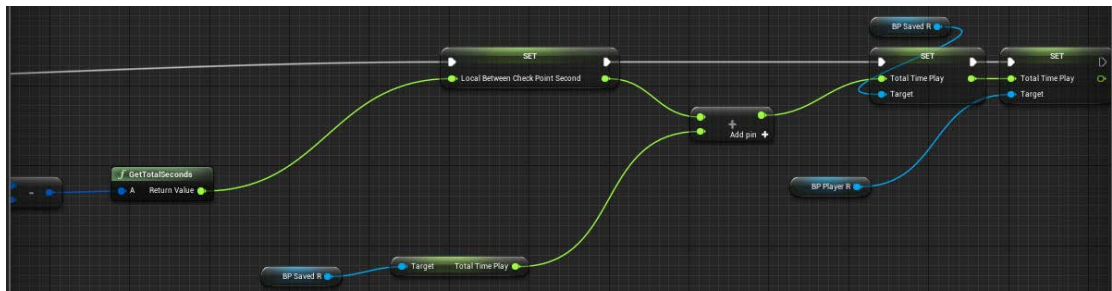
Function Save Game (Part 2)



The diagram illustrates a Blueprint visual scripting logic for a 'Player Equipped' event. It consists of several interconnected nodes and variables:

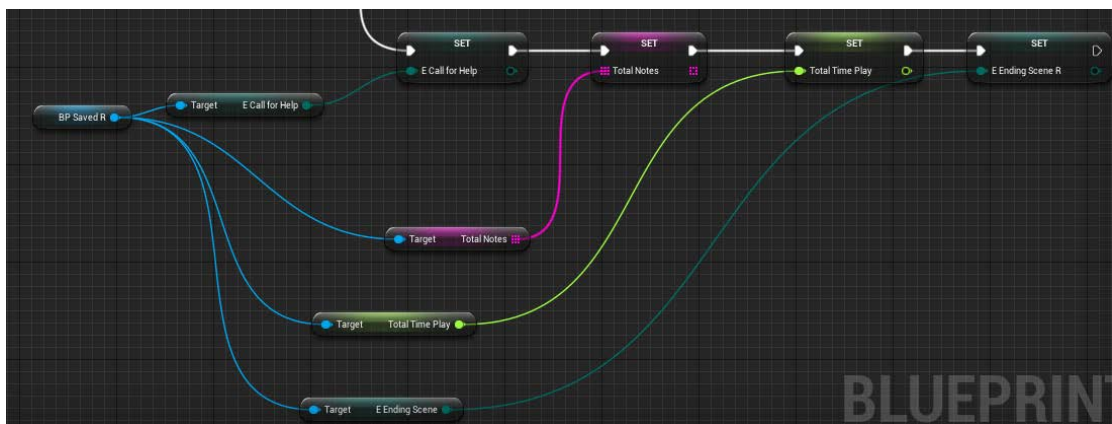
- Variables:**
 - BP Saved R** (Reference to Saved Game)
 - BP Player R** (Reference to Player)
- Event Nodes:**
 - E Player Equipped Stated** (Event triggered when player is equipped)
 - E Player Equipped State R** (Event triggered when player's equipped state changes)
 - E Have Equipment Stated** (Event triggered when equipment is added/removed)
 - E Have Equipment State R** (Event triggered when equipment state changes)
 - E Call for Help** (Event triggered when a call for help is made)
- Logic Nodes:**
 - SET** (Set Variable) nodes used to update **BP Saved R** and **BP Player R**.
 - Is Equipped** (Boolean check) node.
 - Equipped** (Boolean check) node.
- Connections:**
 - E Player Equipped Stated** connects to **SET** (BP Saved R) and **SET** (BP Player R).
 - E Player Equipped State R** connects to **SET** (BP Saved R) and **SET** (BP Player R).
 - E Have Equipment Stated** connects to **SET** (BP Saved R) and **SET** (BP Player R).
 - E Have Equipment State R** connects to **SET** (BP Saved R) and **SET** (BP Player R).
 - E Call for Help** connects to **SET** (BP Saved R) and **SET** (BP Player R).
 - Is Equipped** and **Equipped** nodes are connected to the **SET** (BP Player R) node.

Function Save Game (Part 6)

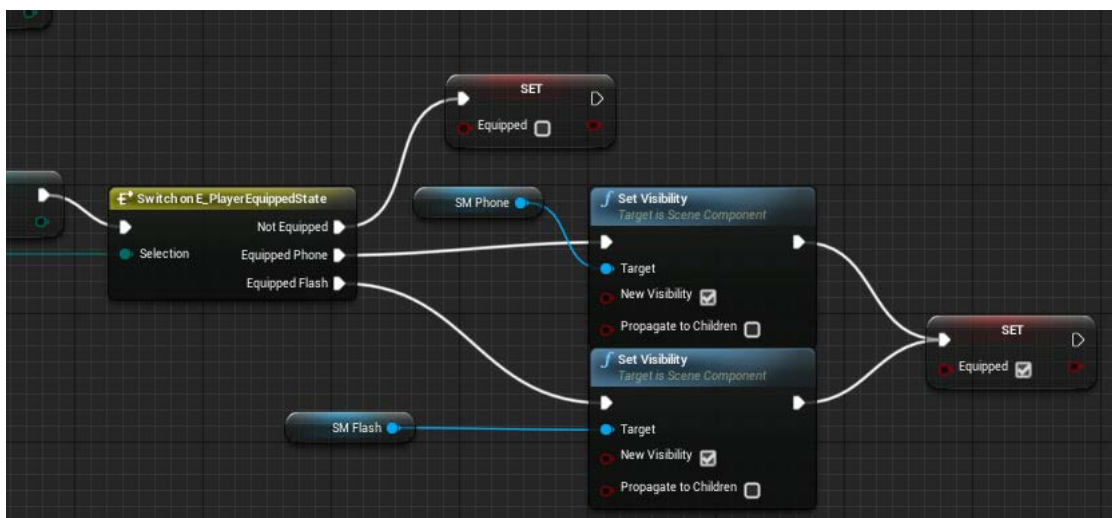


[illegible]

Function Load Game (Part 4)

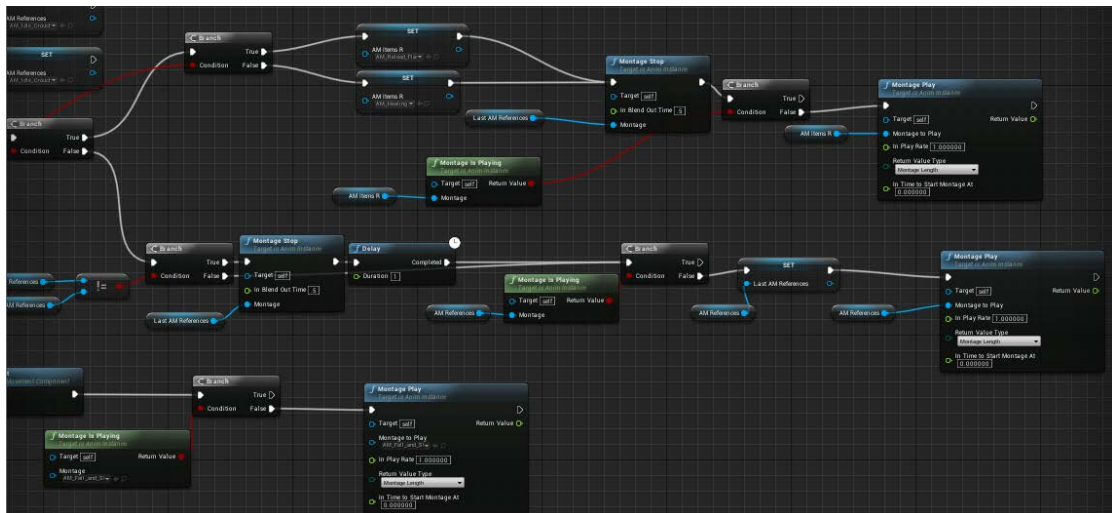


Function Load Game (Part 5)

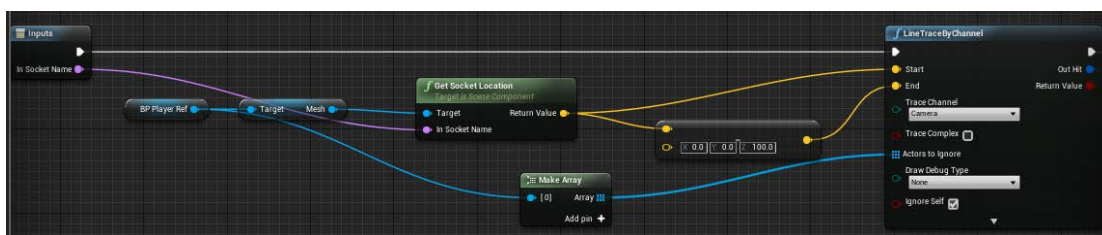


[illegible][illegible]

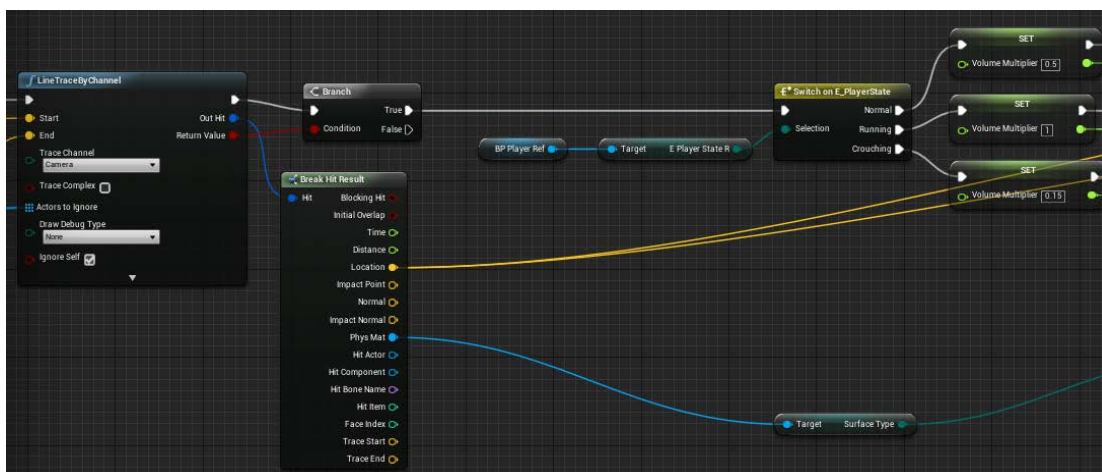
Animation Anim (Player Part 4)



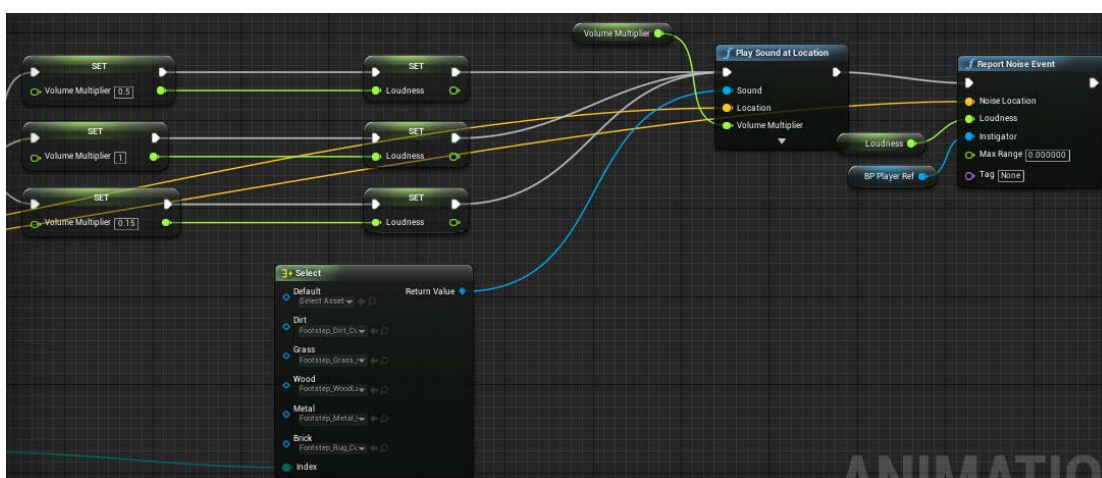
Macro Footstep (Part 1)



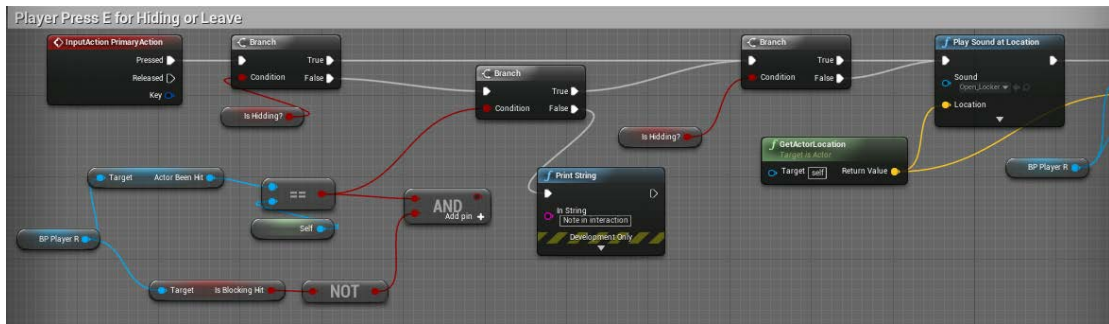
Macro Footstep (Part 2)



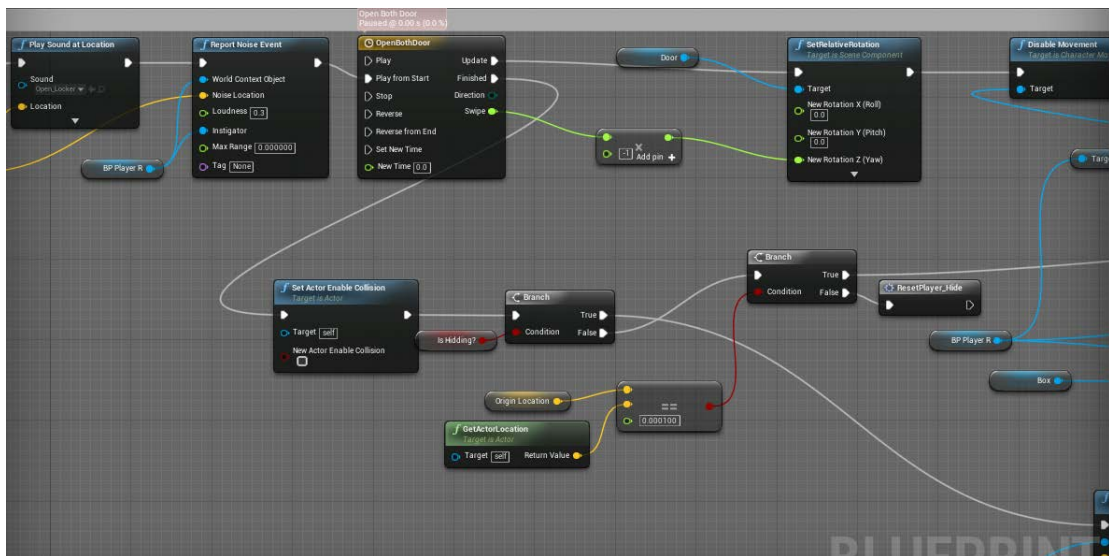
Macro Footstep (Part 3)



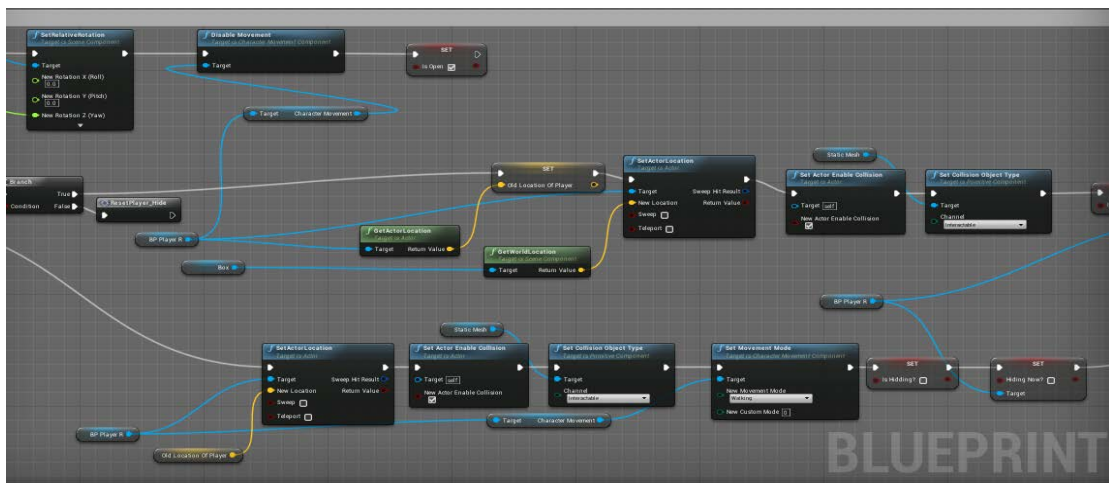
Hide Event (Part 1)



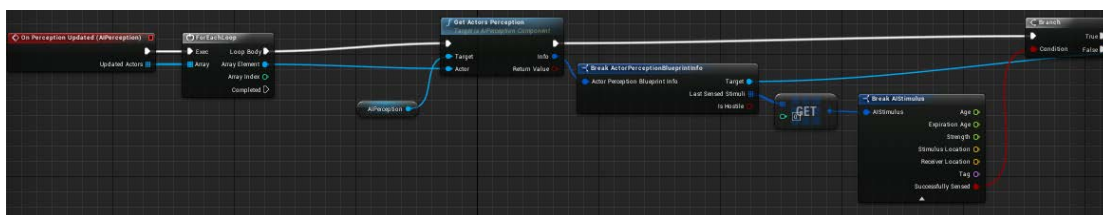
Hide Event (Part 2)



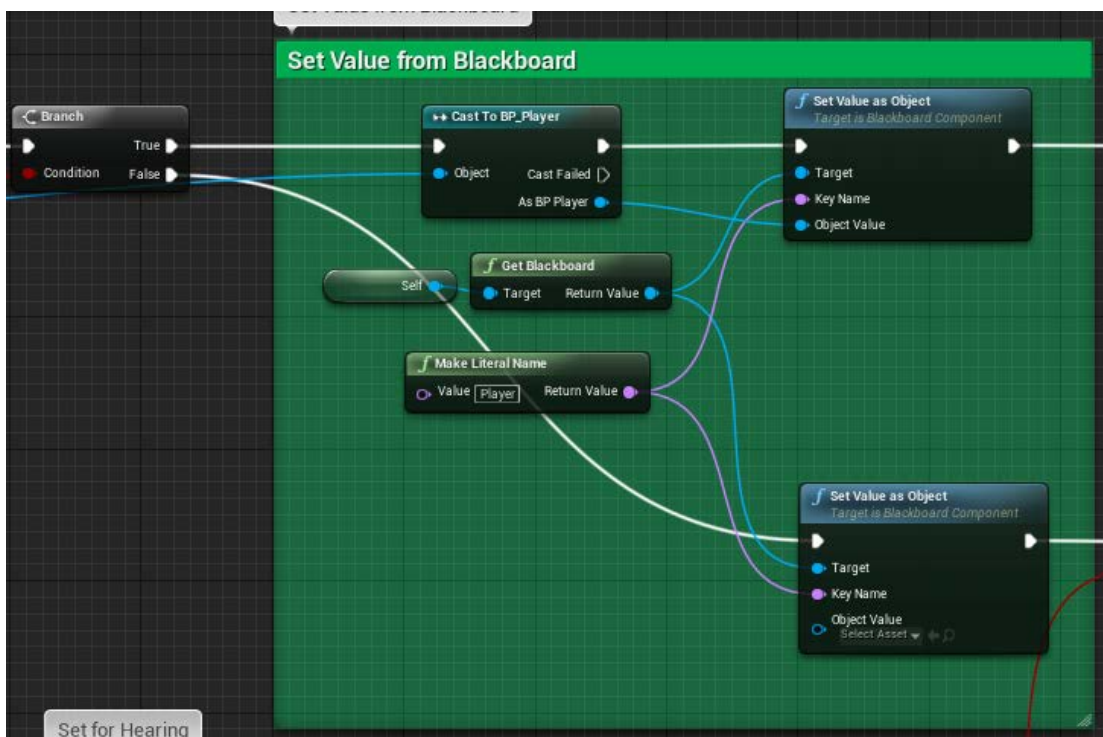
Hide Event (Part 3)



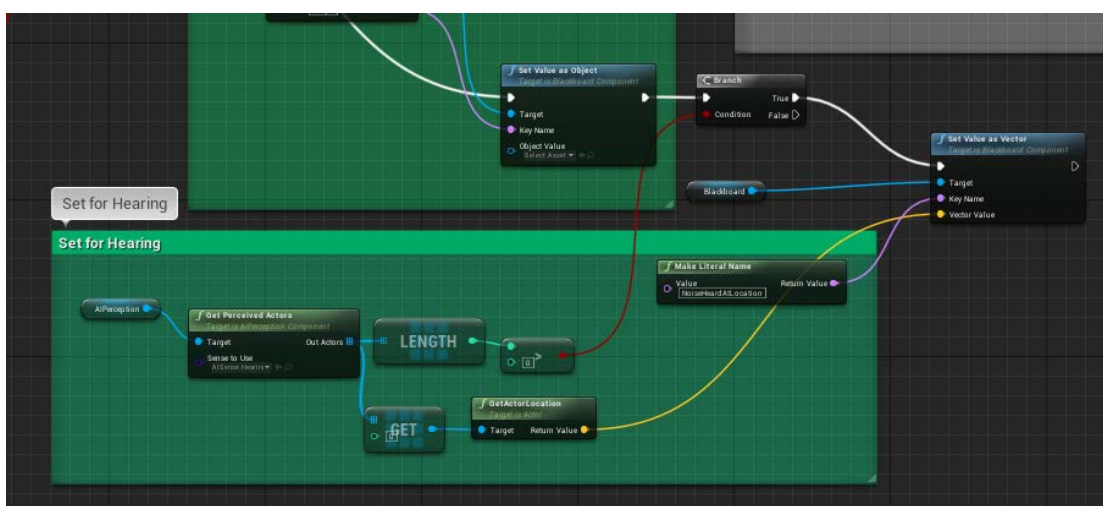
Perception Event (Part 1)



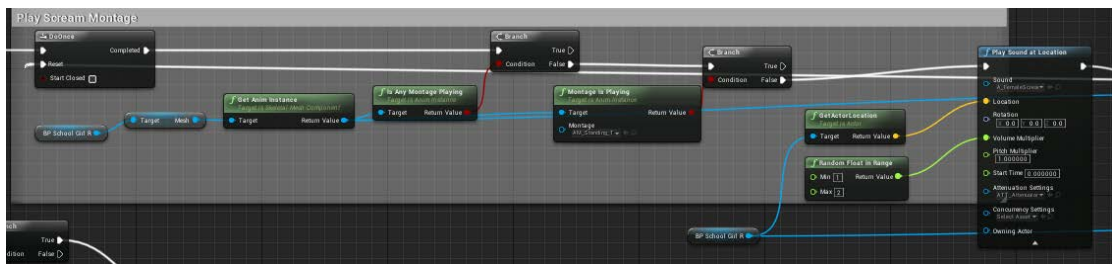
Perception Event (Part 2)



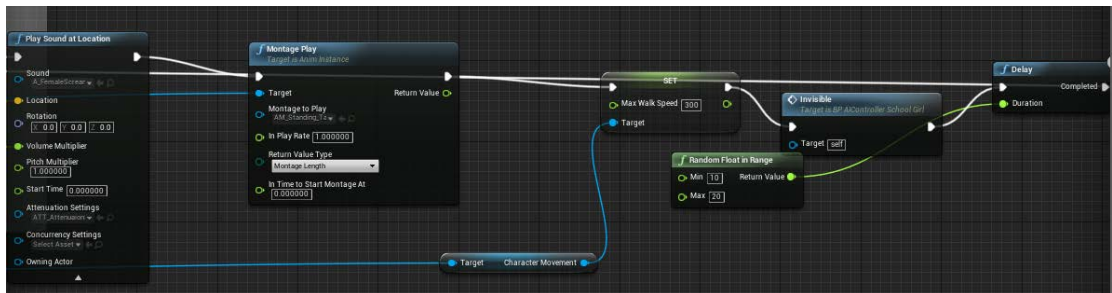
Perception Event (Part 3)



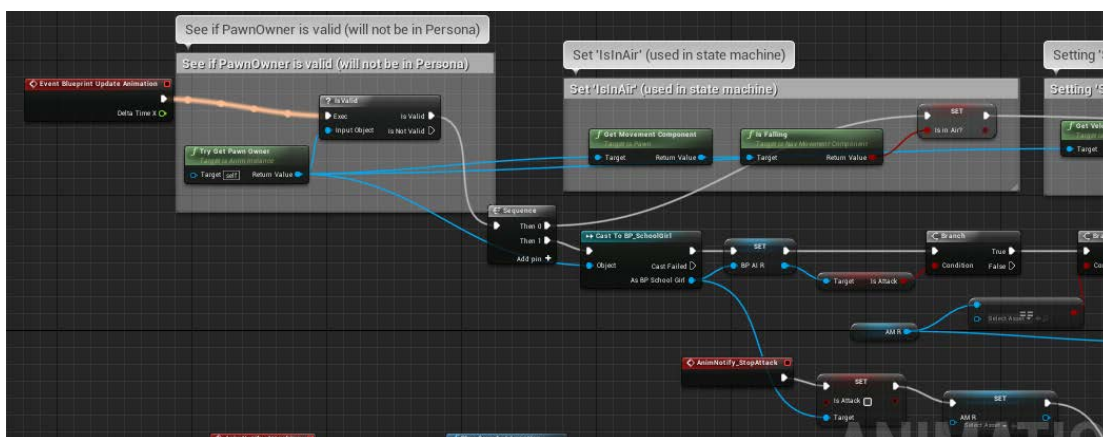
Perception Event (Part 4)



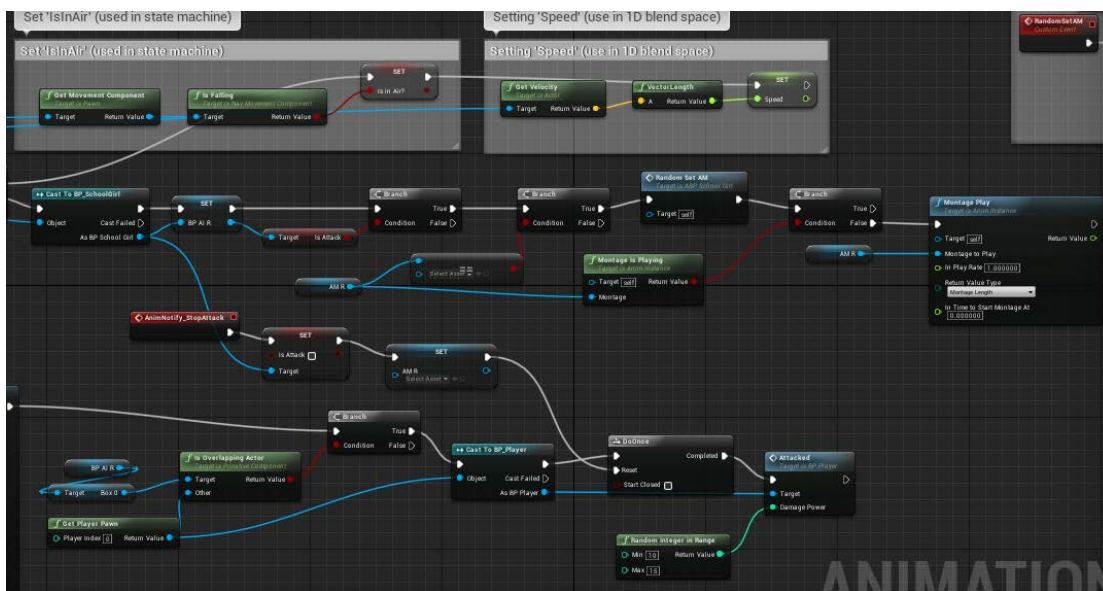
Perception Event (Part 5)



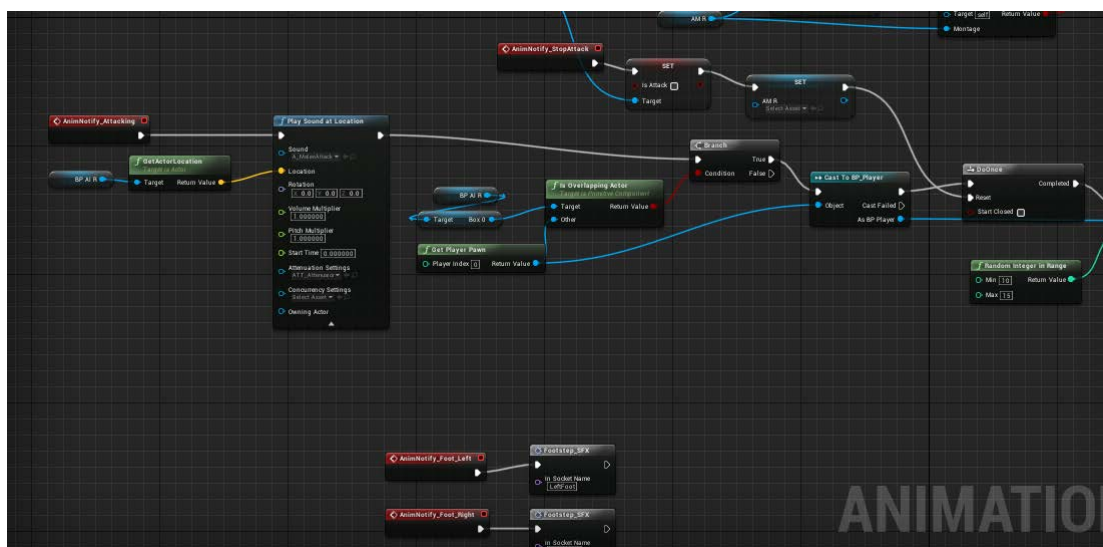
Animation Anim (Enemy Part 1)



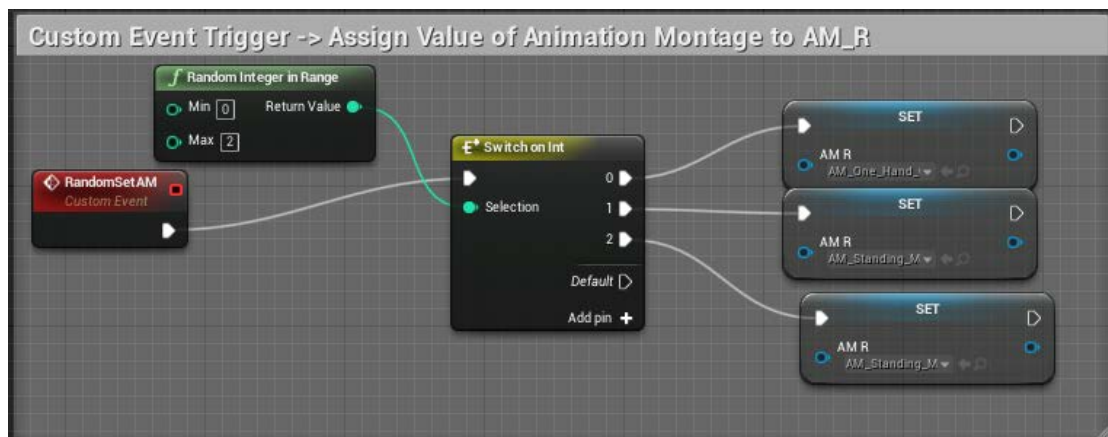
Animation Anim (Enemy Part 2)



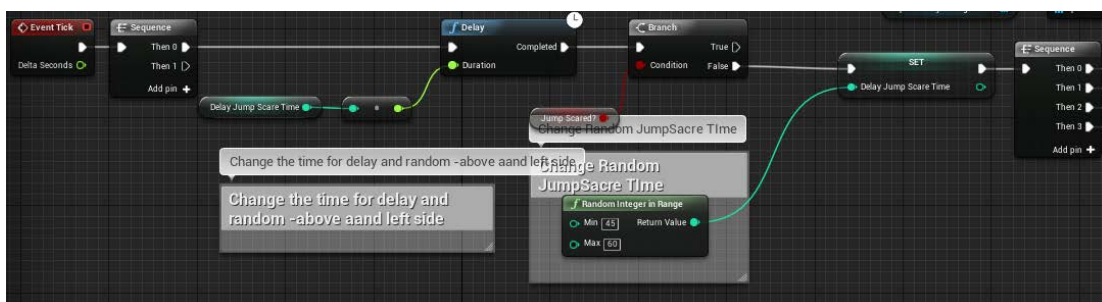
Animation Anim (Enemy Part 3)



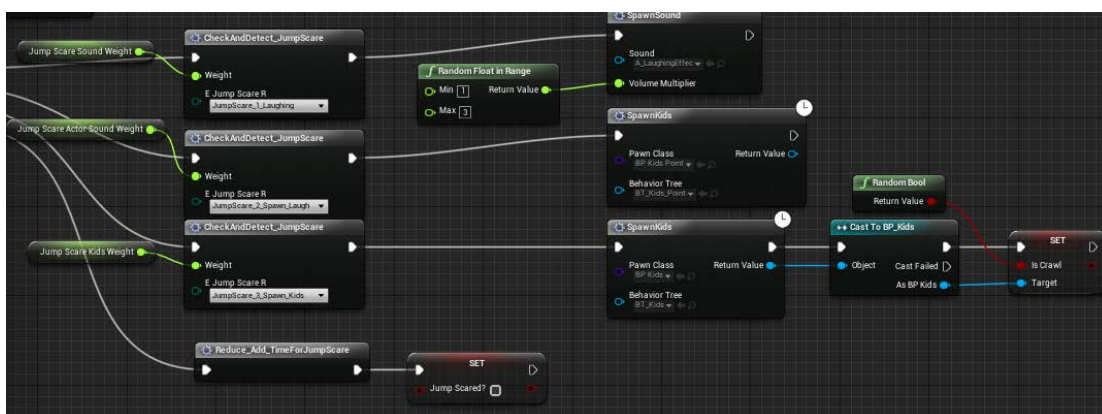
Animation Anim (Enemy Part 4)



Random Jump Score (Part 1)



Random Jump Score (Part 2)



LOGBOOK