

2D GAME-BASED AWARENESS OF CHILD
ABDUCTION FOR PUBLIC :
FOUND YOU

CELIN CHUA WAN JIN

Bachelor of Computer Science
(Graphics and Multimedia Technology)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : CELIN CHUA WAN JIN

Date of Birth

Title : FOUND YOU

Academic Session : 2022/2023

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

(Supervisor's Signature)

New IC/Passport
Number Date: 12 FEB
2023

DR. DANAKORN NINCAREAN A/L EH PHON
SENIOR LECTURER

Name of Supervisor
Date:
FACULTY OF COMPUTING
COLLEGE OF COMPUTING & APPLIED SCIENCES
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR,
TEL: 09-4244643 FAX: 09-4244666

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Graphics & Multimedia Technology) with honours.

A handwritten signature in black ink, appearing to read 'Danakorn', is written over a horizontal line.

(Supervisor's Signature)

Full Name : DR. DANAKORN NINCAREAN A/L EH PHON
SENIOR LECTURER
Position : FACULTY OF COMPUTING
COLLEGE OF COMPUTING & APPLIED SCIENCES
Date : UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR,
TEL: 09-4244643 FAX: 09-4244666



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, consisting of a large, stylized initial 'C' followed by a horizontal line and a small flourish.

(Student's Signature)

Full Name : CELIN CHUA WAN JIN

ID Number : CD19071

Date : 12 FEB 2023

2D GAME-BASED AWARENESS OF CHILD
ABDUCTION FOR PUBLIC :
FOUND YOU

CELIN CHUA WAN JIN

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Graphics & Multimedia Technology) with honours

Faculty of Computer Science
UNIVERSITI MALAYSIA PAHANG

12 FEB 2023

ACKNOWLEDGEMENTS

First, praise is to God for all his blessing me and giving me patience and good health along the project.

Next, I am deeply indebted to my beloved supervisor, Dr. Danakorn Nincarean A/L Eh Phon for his guidance, ideas and advice during the project.

Lastly, I would like to thank my family and friend for their support and encouragement during my preparation for my final year of study. I am appreciate for their cooperation on completing my project by giving support.

ABSTRAK

Situasi penculikan dan pemerdagangan kanak-kanak adalah serius dan perhatian orang ramai adalah rendah. Di Amerika Syarikat, Britain, Australia, Sepanyol dan negara lain, ratusan ribu kanak-kanak bawah umur dan dewasa hilang setiap tahun. Walaupun pemerdagangan kanak-kanak tidak seteruk di Malaysia seperti di negara lain, masih terdapat ratusan kes di Malaysia setiap tahun. Oleh itu kesedaran berasaskan permainan 2D tentang penculikan kanak-kanak bertajuk Found You telah diterbitkan. Permainan Found You boleh membolehkan pemain mempelajari cara mengelak daripada diculik, mempelajari teknik yang akan digunakan oleh penculik apabila mereka cuba menculik dan meningkatkan kesedaran dan pengetahuan tentang penculikan kanak-kanak dalam kalangan pemain dan orang ramai. Objektif projek adalah untuk menyemak aplikasi permainan penculikan kanak-kanak sedia ada, mereka bentuk dan membangunkan aplikasi permainan kesedaran penculikan kanak-kanak 2D, dan menilai kefungsiannya aplikasi permainan yang dibangunkan. Semasa analisis aplikasi permainan sedia ada, didapati aplikasi permainan 2D tentang penculikan kanak-kanak adalah kurang. Projek ini akan dilaksanakan menggunakan Model Pekeliling dengan 4 fasa iaitu Planning and Design, Implementation, Playtest dan Evaluate. Bentuk Ujian Penerimaan Kebolegunaan (UAT) dan Ujian Penerimaan Fungsian (FAT) telah dicipta dan diuji oleh 15 penguji untuk memastikan aplikasi permainan memenuhi jangkaan penguji, permainan memenuhi objektifnya dan berfungsi dengan baik. Dari borang FAT semua keputusan adalah 100% bermakna semua fungsi dalam projek berfungsi seperti yang diharapkan. Untuk borang UAT, 13 daripada 15 penguji (86.7%) bersetuju bahawa projek itu menarik dan mudah difahami. Kesemua 15 penguji bersetuju bahawa projek itu dapat mengajar pemain tentang teknik untuk melindungi diri mereka daripada diculik dan mengetahui teknik atau tingkah laku yang akan digunakan oleh penculik. Selain itu, 14 daripada 15 penguji (93.3%) bersetuju bahawa projek itu boleh meningkatkan kesedaran tentang penculikan kanak-kanak di kalangan pemain.

ABSTRACT

The circumstances of child abduction and trafficking cases are serious and the public attention is low. In the United States, Britain, Australia, Spain and other countries, hundreds of thousands of minors and adults go missing every year. Although child trafficking is not as severe in Malaysia as in other countries, there are still hundreds of cases in Malaysia every year. Therefore a 2D game-based awareness of child abduction titled Found You has been published. Found You game able to allow players learn how to avoid from being kidnapped, learn the techniques will be used by a kidnapper when they try to kidnap, and increase the awareness and knowledge about the child abduction in players and public. The objectives of the project is to study the existing child abduction game application, design and develop a 2D child abduction wareness game application, and evaluate the functionality of the developed game application. During the analyzation on existing game application, it can be found that the 2D game application about child abduction is less. This project will be implement using Spiral Model with 4 phase which are Planning and Design, Implement, Playtest, and Evaluate. A Usability Acceptance Test (UAT) and Functional Acceptance Test (FAT) form is created and tested by 15 testers to ensure the game application is satisfied with tester expectation, the game is meet the objectives, and it is function well. From the FAT form all the results are 100% which means all the functions in the project are worked as expectation. For the UAT form, 13 out of 15 testers (86.7%) agree that the project is interesting and easy to understand. All the 15 testers are agree that the project is able to teach player about the techniques of protect themselves from being kidnapped and know the techniques or treats that will be used by kidnappers. Besides, 14 out of 15 testers (93.3%) agree that the project is able to increase the awareness of child abduction in player.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1-2
1.2 Problem Statements	2-3
1.3 Objectives	3
1.4 Scopes	4
1.3 Significant of the project	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Introduction	5
2.2 Review of Existing System	5
2.2.1 Toby : The Secret Mine	6-9
2.2.2 LIMBO	10-13
2.2.3 The Final Station	13-19
2.3 Comparison Table of the Existing Game Application	19-21

2.4	Chapter Summary	22
CHAPTER 3 METHODOLOGY		23
3.1	Introduction	23
3.2	Spiral Model	24
3.2.1	Planning & Design Phase	24-25
3.2.2	Implement Phase	25
3.2.3	Play Test Phase	25
3.2.4	Evaluate Phase	25
3.3	Spiral Model in the Development Process of the Project	25-27
3.4	Project Requirements	27-28
3.4.1	Functional Requirements	28
3.4.2	Non-Functional Requirements	29
3.4.3	Constraints & Limitations	30
3.5	User Requirements	30-33
3.6	Proposed Design	34
3.6.1	Flowchart	34-36
3.6.2	Context Diagram	37
3.6.3	Use Case Diagram	38
3.6.4	Activity Diagram	39
3.6.5	Game Content	40-42
3.6.6	Game Mechanic	42-43
3.6.7	Game Technology	43
3.6.8	Storyboard	44-48
3.7	Data Design	49

3.7.1	ERD Diagram	49
3.7.2	Data Dictionary	49
3.8	Interface Design	50
3.8.1	Main Menu	50
3.8.2	Game Interface	50-51
3.8.3	Pause Menu	51
3.8.4	Lose Scene	52
3.8.5	Win Scene	52
3.9	Testing Plan	53-55
3.10	Gantt Chart	56
3.11	Potential Use of Proposed Solution	57
CHAPTER 4 RESULT AND DISCUSSION		58
4.1	Introduction	58
4.2	Implementation Phase	58
4.2.1	Game Assets (adobe Photoshop)	58-61
4.2.2	Main Menu	62-64
4.2.3	Pause Menu	64-66
4.2.4	Lose Panel	66-68
4.2.5	Highscore Panel	68-70
4.2.6	Game Beginning Instruction	70-74
4.2.7	Game User Interface	75
4.2.8	Game Level Interface	76-125
4.2.9	Win Scene	126-127
4.2.10	Player Controller	128
4.2.11	Camera	129

4.2.12	Dialogue System	129-131
4.2.13	Open Furniture	131-132
4.2.14	Score	132
4.2.15	Enemy Behavior	133-134
4.2.16	Bandage	134
4.3	Evaluattion Phase	135
4.3.1	Result of Functional and User Acceptance Test	135-153
4.4	Chapter Summary	153
CHAPTER 5 CONCLUSION		154
5.1	Introduction	154
5.2	Limitation	155
5.3	Future Work	155
RECEARNCES		156-160
APPENDIX A - FAT & UAT PROOF		161-163
APPENDIX B - C# SCRIPTING		164-191

LIST OF TABLES

Table 2.1	Information and Minimum Requirements of Toby: The Secret Mine	7
Table 2.2	Information and Minimum Requirements of LIMBO	11
Table 2.3	Information and Minimum Requirements of The Final Station	14
Table 2.4	Comparison of the Existing Game Applications	20-21
Table 3.1	Task Distribution for Spiral Model in the Project	27
Table 3.2	Data Responses from Google Form	31-33
Table 3.3	Storyboard of the Game	44-48
Table 3.4	Data Dictionary of the Player Entity	49
Table 3.5	Data Dictionary of the System Entity	49
Table 3.6	UAT (User Acceptance Test) Plan of the Project	53-55
Table 4.1	Result of FAT and UAT From	135-153

LIST OF FIGURES

Figure 2.1	Toby : The Secret Mine	6
Figure 2.2	The Main Menu UI of Toby: The Secret Mine	8
Figure 2.3	The Loading Page UI of Toby: The Secret Mine	8
Figure 2.4	Game UI of Toby: The Secret Mine	9
Figure 2.5	Puzzle UI of Toby: The Secret Mine	9
Figure 2.6	LIMBO	10
Figure 2.7	The Main Menu UI of LIMBO	12
Figure 2.8	The Setting UI of LIMBO	12
Figure 2.9	Game UI of LIMBO	13
Figure 2.10	The Final Station	13
Figure 2.11	The Main Menu UI of The Final Station	15
Figure 2.12	The Setting UI of The Final Station	15
Figure 2.13	Game UI of The Final Station	16
Figure 2.14	The Dialogue and Notes in The Final Station	16
Figure 2.15	The Shop UI of The Final Station	17
Figure 2.16	The Battery UI in Train Interior part of The Final Station	17
Figure 2.17	The Message UI in Train Interior part of The Final Station	18
Figure 2.18	The Map UI in Train Interior part of The Final Station	18
Figure 2.19	The Reserve UI in Train Interior part of The Final Station	19
Figure 3.1	Phase of the Software Development Lifecycle (SDLC)	23
Figure 3.2	Simple Diagram of Spiral Model	24
Figure 3.3	Detailed Diagram of Spiral Model	26
Figure 3.4	Flowchart of the Main Menu	34
Figure 3.5	Flowchart of the Game Play in a Level	35
Figure 3.6	Flowchart of the Pause Menu	36
Figure 3.7	Context Diagram of the Project	37
Figure 3.8	Use Case Diagram of the Project	38
Figure 3.9	Activity Diagram of the Project	39
Figure 3.10	ERD Diagram of the Project	49
Figure 3.11	Interface Design of the Main Menu	50
Figure 3.12	Interface Design of the Game Interface	51
Figure 3.13	Interface Design of the Pause Menu	51
Figure 3.14	Interface Design of the Lose Scene	52

Figure 3.15	Interface Design of the Win Scene	52
Figure 3.16	Gantt Chart of the Project	56
Figure 4.1	Interface of the Adobe Photoshop	59
Figure 4.2	Draw outline of game asset	59
Figure 4.3	Color the game asset	60
Figure 4.4	Export game assets as PNG	60
Figure 4.5	Created game assets	61
Figure 4.6	Create Characters, Enemies and Animation	61
Figure 4.7	Main Menu Interface	62
Figure 4.8	Inspector of the Main Menu Buttons	63
Figure 4.9	ChangeScene_highscore Script in Main Menu	64
Figure 4.10	Pause Menu Interface	64
Figure 4.11	Inspector of the Pause Menu Buttons	65
Figure 4.12	ChangeScene Script in Main Menu	66
Figure 4.13	Lose Panel Interface	66
Figure 4.14	Inspector of the Lose Panel Buttons	67
Figure 4.15	ChangeScene Script in Lose Panel	68
Figure 4.16	Highscore Panel Interface	68
Figure 4.17	Inspector of the Hoghscore Panel Buttons	69
Figure 4.18	ChangeScene_highscore Script in Highscore Panel	70
Figure 4.19	Game Beginning Instruction 1 Interface	71
Figure 4.20	Game Beginning Instruction 2 Interface	71
Figure 4.21	Game Beginning Instruction 3 Interface	72
Figure 4.22	Inspector of the Game Beginning Instruction 1 Buttons	72
Figure 4.23	Inspector of the Game Beginning Instruction 2 Buttons	73
Figure 4.24	Inspector of the Game Beginning Instruction 3 Buttons	73
Figure 4.25	ChangeScene Script in Lose Panel Beginning Instruction	74
Figure 4.26	Game User Interface	75
Figure 4.27	Game User Interface Development in Unity	75
Figure 4.28	Game Environment Design of Level 1	76
Figure 4.29	Level 1Anne's House Development	76
Figure 4.30	Timeline of the Game Begin in Level 1	77
Figure 4.31	Tutorial 1 Player Movement Development	77
Figure 4.32	lv1gamebegin Script in Tutorial 1	78
Figure 4.33	Interfade of the tutorial 1 in Level 1	78

Figure 4.34	lv1tutorial1To2 Script in Tutorial 1	79
Figure 4.35	Interface of the Tutorial 2 in Level 1	80
Figure 4.36	Tutorial 2 Open Door Development	80
Figure 4.37	lv1speciallockedDoor Script in Tutorial 2	81
Figure 4.38	Interface of the Tutorial 3 in Level 1	81
Figure 4.39	Tutorial 3 Open Furniture Development	82
Figure 4.40	lv1TutorialopenFurniture Script in Tutorial 3	82
Figure 4.41	Interface of the Read Message / Note Tutorial in Level 1	83
Figure 4.42	Message Leave by Anne's mom in Level 1	83
Figure 4.43	Clue 1 from Emily Interface in Level 1	84
Figure 4.44	Clue 1 from Emily Development in Level 1	84
Figure 4.45	Poster About "Don't Open Doors To Starnger" Interface in Level 1	85
Figure 4.46	Poster About "Don't Open Doors To Starnger" Development in Level 1	85
Figure 4.47	Level 1 Lesson Timeline Interface	86
Figure 4.48	Level 1 Lesson Timeline Interface Development	86
Figure 4.49	starttimeline Script in Lesson Detector Level 1	87
Figure 4.50	Level 1 Selection Panel Interface	87
Figure 4.51	Level 1 Selection Panel Interface Development	88
Figure 4.52	theSelectionScript Script for Selection Panel in Level 1	88
Figure 4.53	Shooting Tutorial Timeline Interface in Level 1	89
Figure 4.54	Shooting Tutorial Timeline Development in Level 1	89
Figure 4.55	Resident Dialogue in Level 1	90
Figure 4.56	Dialogue of the Resident in Level 1	90
Figure 4.57	Map Drawn by Emily in Level 1	91
Figure 4.58	Map Drawn by Emily Interface Development in Level 1	91
Figure 4.59	"Kidnapping Cases In Malaysia" News Interface in Level 1	92
Figure 4.60	"Kidnapping Cases In Malaysia" News Interface Development in Level 1	92
Figure 4.61	Clue 2 From Emily in Level 1	93
Figure 4.62	Checkpoint Development in Level 1	93
Figure 4.63	Game Environment Ddesign of Level 2	94
Figure 4.64	Clue 3 From Emily in Level 2	94
Figure 4.65	Resident 1 Dialogue in Level 2	95
Figure 4.66	Dialogue of the Resident 1 in Level 2	95
Figure 4.67	Resident 2 Dialogue in Level 2	96
Figure 4.68	Dialogue of the Resident 2 in Level 2	96

Figure 4.69	Resident3 (Worker 1) Dialogue in Level 2	97
Figure 4.70	First Dialogue of the Resident 3 (Worker 1) in Level 2	97
Figure 4.71	Second Dialogue of the Resident 3 (Worker 1) in Level 2	97
Figure 4.72	Resident3 (Worker 2) Dialogue in Level 2	98
Figure 4.73	First Dialogue of the Resident 3 (Worker 2) in Level 2	98
Figure 4.74	Second Dialogue of the Resident 3 (Worker 2) in Level 2	98
Figure 4.75	Dialogue of Worker 1 & 2 Development in Level 2	99
Figure 4.76	Resident 4 Dialogue in Level 2	99
Figure 4.77	Dialogue of the Resident 4 in Level 2	100
Figure 4.78	Keypad Code in Level 2	100
Figure 4.79	Small Code Hint Interface in Level 2	101
Figure 4.80	Keypad Panel Interface in Level 2	101
Figure 4.81	PushtheBtn Script for Keypad Button in Level 2	102
Figure 4.82	DigitDisplay Script for Keypad in Level 2	102
Figure 4.83	"Run and Get Help" Message in Level 2	103
Figure 4.84	Clue 4 From Emily in Level 4	103
Figure 4.85	Poster about Do Not Walk In Less Crowded Place in Level 2	104
Figure 4.86	Respawn Point in Level 2	104
Figure 4.87	checkpointPlayerlv2 Script in Level 2	105
Figure 4.88	Kidnapper harassment Timeline in Level 2	105
Figure 4.89	Kidnapper harassment Timeline Collider in Level 2	106
Figure 4.90	Kidnapper Chase in Level 2	106
Figure 4.91	lv2kidnapperFoolowPlayer Script in Level 2	107
Figure 4.92	Close Door Instruction in Level 2	107
Figure 4.93	Close Door Instruction Collider in Level 2	108
Figure 4.94	Selection Panel in Level 2	108
Figure 4.95	starttimelineLV2script Script in Level 2	109
Figure 4.96	Win Timeline in Level 2	109
Figure 4.97	Checkpoint in Level 2	110
Figure 4.98	Game Environment Design of Level 3	110
Figure 4.99	Emily Shoe Panel in Level 3	111
Figure 4.1.1	Instruction of Emily's Shoe Level 3	111
Figure 4.1.2	Picture About "Do Not Ride Stranger's Car" in Level 3	112
Figure 4.1.3	News About "Kidnapping Fact in Melaka" in Level 3	112
Figure 4.1.4	Picture About "Do Not Take Anything Frome a Stranger" in Level 3	113

Figure 4.1.5	Kidnapper Timeline Detector in Level 3	113
Figure 4.1.6	Kidnapper Timeline Part 1 in Level 3	114
Figure 4.1.7	Selection Panel 1 in Level 3	114
Figure 4.1.8	Kidnapper Timeline Part 2 in Level 3	115
Figure 4.1.9	Selection Panel 2 in Level 3	115
Figure 4.1.10	Checkpoint in Level 3	116
Figure 4.1.11	Game Environment Design of Level 4	116
Figure 4.1.12	Find Emily Timeline in Level 4	117
Figure 4.1.13	Meet Police Timeline in Level 4	117
Figure 4.1.14	Picture About Do Anything To Escape in Level 4	118
Figure 4.1.15	Picture About Call Police If Meet The Kidnapper in Level 4	118
Figure 4.1.16	Kidnapper Talk Timeline in Level 4	119
Figure 4.1.17	Kidnapper Chase in Level 4	119
Figure 4.1.18	Kidnapper Chasing Waypoints in Level 4	120
Figure 4.1.19	lv4KidnapperChase Script in Level 4	120
Figure 4.1.20	Struggle Panel in Level 4	121
Figure 4.1.21	lv4Struggle Script in Level 4	121
Figure 4.1.22	Selection 1 Panel in Level 4	122
Figure 4.1.23	Selection 2 Panel in Level 4	123
Figure 4.1.24	Selection 3 Panel in Level 4	123
Figure 4.1.25	Selection 4 Panel in Level 4	124
Figure 4.1.26	Win Panel in Level 4	124
Figure 4.1.27	Lose Panel in Level 4	125
Figure 4.1.28	Game Ending Timeline in Level 4	125
Figure 4.1.29	Win Scene Interface	126
Figure 4.1.30	Inspector of the Win Scene Buttons	127
Figure 4.1.31	highscore Script in Win Scene	127
Figure 4.1.32	PlayerController Script	128
Figure 4.1.33	Shooting Script	128
Figure 4.1.34	CameraFollow Script	129
Figure 4.1.35	DialogueTrigger Script Location	129
Figure 4.1.36	Dialogue Script Location	130
Figure 4.1.37	DialogueTrigger Script	130
Figure 4.1.38	Dialogue Script	131
Figure 4.1.39	Openfurniture Script Location	131

Figure 4.1.40 Openfurniture Script	132
Figure 4.1.41 ScoreCounter Script	132
Figure 4.1.42 EnemyPatrol Script	133
Figure 4.1.43 EnemyHealtManage Script	133
Figure 4.1.44 EnemyControl Script	134
Figure 4.1.45 collideBandage Script	134

CHAPTER 1

INTRODUCTION

1.1 Introduction

About 1,000 bodies remain unidentified a year later, according to the National Institute of Justice. Statistics on kidnappings and missing persons show that a large number of bodies remain anonymous for long periods of time. In this case, time is the mortal enemy. The longer a body remains unidentified, the harder it is to recover any actual information about the person's identity. (National Missing and Unidentified Person System, 2022)

Annual statistics on child abductions show that children are the most common targets of kidnappers. While most of the children who went missing in 2020 were later found alive, the data revealed a shocking truth - kidnappers tend to go after the most vulnerable. (FBI, 2020)

The missing children report in every year for certain countries : in Australia, an estimated 20,000 children (Australian Federal Police, National Coordination Centre), in Canada, an estimated 45,288 children (Government of Canada, Canada's Missing – 2015 Fast Fact Sheet), in Germany, an estimated 100,000 children (Initiative Vermisste Kinder), in India, an estimated 96,000 children (Bachpan Bachao Andolan, Missing Children of India), in Jamaica, an estimated 1,984 children (Jamaica's Office of Children's Registry, 2015), in Russia, an estimated 45,000 children (Interview with Pavel Astakhov MIA "Russia Today", Apr. 4, 2016), in Spain, an estimated 20,000 children (Spain Joins EU Hotline for Missing Children, Sep. 22, 2010), in the United Kingdom, an estimated 112,853 children (National Crime Agency, UK Missing Persons Bureau) and in the United States, an estimated 460,000 children (Federal Bureau of Investigation, NCIC)

From 2012 to 2017, there were 482 cases of missing children under the age of 12 in Malaysia, and statistics for 2017 show that an average of 4 children go missing every day. While the figures include runaway teens, the statistics also include children kidnapped for ransom, child trafficking or sexual exploitation — some of whom never come back alive or even discovered. (Vignes, 2019)

However, this is only a microcosm of the problem. In many countries, statistics on missing children are not even available. Unfortunately, even available statistics may be inaccurate due to under-reporting or under-accreditation, inflation, inaccurate case information database entry Correct, records are deleted after the case is closed.

Therefore, the project aims to develop a 2D game to increase awareness among the public including children about child abduction. This game educates people to be more vigilant about child abduction cases, and can also educate the younger generation on how to avoid and protect themselves from being abducted.

1.2 Problem Statements

The most common problem is the lack of awareness among the public, parents and children about child abduction. While there are many articles, news, and advertisements about child trafficking, views and engagement rates are low. Even if thousands of missing children's notices are posted on the streets, they may not get the attention and help of the public. The public must be vigilant and take child abduction cases seriously. Public vigilance is the best weapon to limit the actions of kidnapping suspects, or even deter anyone from planning any kidnapping.

Many children go missing because strangers take them away. Family members didn't even think it was kidnapping at first, but legally it was. This happens just because of a lack of awareness about child abduction. When children go missing due to kidnapping, it is especially important to find them quickly. While children are sometimes discovered years or decades after the fact, it becomes more difficult to reintegrate them into the family.

Second, children have less information and knowledge about child abduction. Parents are not playing their part in teaching their children how to protect and avoid kidnapping. Children must have basic knowledge of kidnapping, such as situations with high chances of kidnapping and techniques commonly used by kidnappers. Children also need to learn how to protect themselves if they are kidnapped.

In addition, there are very few game apps with the theme of kidnapping children. It is easy to get bored reading long articles and news about child abductions. As such, games are a promising awareness-raising tool that can stimulate and enhance players' interest in child abduction and require only a small investment of time.

1.3 Objectives

1. To study the existing child abduction awareness game applications.
2. To design and develop a 2D child abduction awareness game application.
3. To evaluate the functionality of the developed child abduction awareness game application.

1.4 Scopes

A. User scope :

- Players that are aged 5 and above.
- Players that use PC devices.

B. System scope :

- Single player game
- No internet is needed.
- Control the character by pressing the arrows button to move left, right, up, and down, and space bar for actions.

C. Development scope :

- Graphics : 2D Hand-drawing style, the main colour is black, white and grey (dark colour) to create a mysterious atmosphere. Smooth animation and narrative will provide.
- Audio : The background music will use the soft and slow BGM and sound effect will provide an increased gaming experience.
- Using Photoshop, Unity Software

1.5 Significant of the project

Throughout the game, if the public does not take child kidnapping seriously, players will become aware of the impact. In the game, NPCs who are public figures show indifference to missing children, and eventually the number of missing children increases, and even some kidnapped children die.

The game will allow players to interact with unfamiliar NPCs, and choices will lead players to different endings. Thus, players can learn which are the appropriate actions to avoid being kidnapped by strangers. At the end of the game, players will understand what can be done to avoid their kidnapping and raise player awareness about child abduction.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter is to study and review existing 2D game applications in the adventure game genre. In addition, this chapter also studies the differences, advantages and disadvantages of existing game applications through comparison, and the comparison results will be used as a reference for the features of game applications in this project.

2.2 Review of Existing System

The review focuses on existing 2D adventure game applications. Existing game applications that have been selected for review are **Toby: The Secret Mine**, **LIMBO**, and **The Final Station**. These game applications have similar gameplay, game genre, and storylines, and are all about characters who have their own mission (find the truth or find friends and sisters) and start their journey.

2.2.1 Toby: The Secret Mine

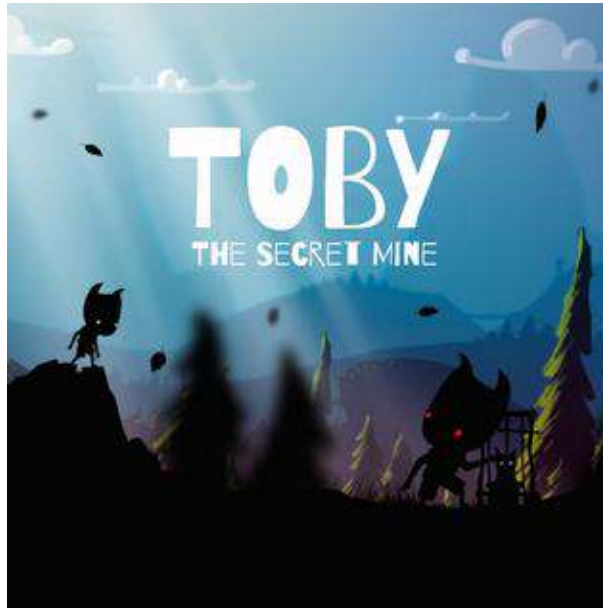


Figure 2.1 Toby: The Secret Mine

Toby: The Secret Mine is a side-scrolling puzzle platformer with black silhouette graphics. The gameplay is very similar to Limbo, but it has a less dark atmosphere and a more colourful world. Players control Toby throughout the game, guiding him through multiple levels in different environments, such as forests, caves, mines, blizzards, and more.

Players must solve puzzles along the way. Players can easily ignore multiple traps unless vigilant. These traps vary by environment, and Blizzard levels are largely focused on timed platforms. The entire level also hides kidnapped friends, adding an element of exploration. The game has a total of 21 different levels.

Based on Table 2.1, the developer of Toby: The Secret Mine is Lukáš Navrátil. The target player of the game is everyone that is 10 years old or older. The game engine used to develop the game is Unity. This game is available on Windows, Linux, OS X, iOS, Android, Xbox One, Wii U, PlayStation 4, and Switch. Toby: The Secret Mine is a single-player puzzle-adventure platformer. The minimum system requirements to run this game are operating system (OS) that is Win 7 or later, processor that is Intel Core

i3 or better, minimum 2GB RAM of memory, graphics is Intel HD 4000 or better, minimum DirectX version 9.0 and a 800MB available storage space.

Table 2.1 Information and Minimum Requirements of Toby: The Secret Mine

TYPE	CONTENT
Developer	Lukáš Navrátil
Target Player	Everyone 10 years old or older
Game Engine	Unity
Platform	Windows, Linux, OS X, iOS, Android, Xbox One, Wii U, PlayStation 4, Switch
Genre	Puzzle, Adventure, Platform
Mode	Single Player
Operating System	Win 7 or later
Processor	Intel Core i3 or better
Memory	2GB RAM
Graphics	Intel HD 4000 or better
DirectX	version 9.0
Storage	800 MB available space

Figure 2.2 is the Main Menu User Interface of the game. There are 3 buttons provided in the Main Menu which are Help, Level and Play buttons. The Help button allows players to view the instruction of the gameplay, Level button allows players to view and replay the level that they want, while Play button allows players to start the game. Besides, on the right top of the Main Menu will show the current number of residents that have been saved by players.



Figure 2.2 The Main Menu UI of Toby: The Secret Mine

Figure 2.3 is the Loading Page User Interface of the game. The Loading Page is used for every navigation between scenes. A smooth animation can be seen in the middle of the Loading Page, and there is a "LOADING" word below the animation to let the player know that the game is loading.



Figure 2.3 The Loading Page UI of Toby: The Secret Mine

Figure 2.4 is a User Interface in the game. Players are required to control the character to save his friends. There is a Pause button at the left top of the UI, which allows players to pause the game. The right top of the UI will show the number of residents that have been saved by players.



Figure 2.4 Game UI of Toby: The Secret Mine

Figure 2.5 is the Puzzle User Interface of the game. This UI will appear once players are exposed to the puzzle mechanism. Players are required to click on the correct symbol to solve the puzzle.



Figure 2.5 Puzzle UI of Toby: The Secret Mine

2.2.2 LIMBO

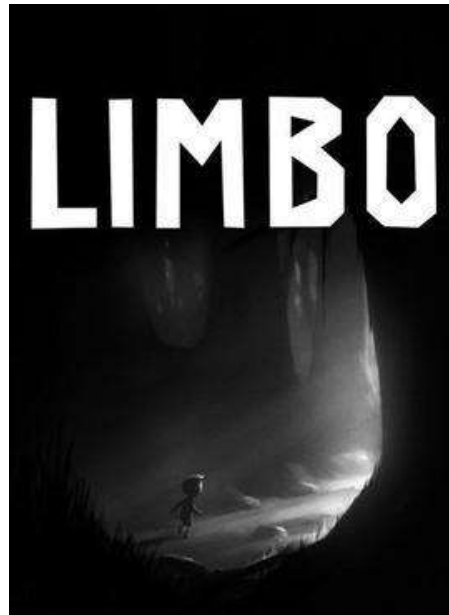


Figure 2.6 LIMBO

Limbo is a puzzle platform video game developed by indie studio Playdead and originally published by Microsoft Game Studios for the Xbox 360. This game is a 2D side-scrolling game that uses a physics system to manage environmental objects and player characters. Players guide a nameless boy through dangerous environments and traps in search of his sister. Developers build the game's puzzles with the expectation that players will fail until the correct solution is found. Playdead calls the playstyle "judgment and death" and uses a horrific image of a dying boy to steer players away from unworkable solutions.

The player controls the boy throughout the game. Like most 2D platformers, boys can run left and right, jump, climb short ledges or go up and down ladders and ropes, as well as push and pull objects. Limbo creates eerie, haunting environments using deep grayscale graphics and minimal ambient sound rendering. The dark visuals also hide many deadly surprises, including environmental and physical hazards, such as deadly bear traps on the forest floor or deadly monsters hiding in the shadows. Among the dangers were glowing worms that would cling to the boy's head, forcing him to move in only one direction until he was killed.

Based on Table 2.2, the developer of LIMBO is Playdead. The target player of the game is a teenager 18 years old or older. The game engine used to develop the game is Box2D. This game is available on Xbox 360, PlayStation 3, Windows, OS X, Linux, Xbox One, PlayStation 4, PlayStation Vita, Nintendo Switch, iOS, and Android. LIMBO is a single-player puzzle-adventure platformer. The minimum system requirements to run this game are operating system (OS) that are Windows XP, Vista, and 7, minimum processor is 2 GHz, minimum 512MB of memory, use of a video card that is 5 years or younger and Shader Model 3.0 for graphics, and DirectX version 9.0c.

Table 2.2 Information and Minimum Requirements of LIMBO

TYPE	CONTENT
Developer	Playdead
Target Player	Teenager 18 years old or older
Game Engine	Box2D
Platform	Xbox 360, PlayStation 3, Windows, OS X, Linux, Xbox One, PlayStation 4, PlayStation Vita, Nintendo Switch, iOS, Android
Genre	Puzzle, Adventure, Platform
Mode	Single Player
Operating System	Windows XP, Vista, 7
Processor	2GHz
Memory	512 MB
Graphics	Video Card 5 years or younger, Shader Model 3.0
DirectX	version 9.0c

Figure 2.7 is the Main Menu User Interface of LIMBO. There are five buttons provided which are Resume, Start New Game, Load Chapter, Settings and Exit. The Resume button allows players continue the game, Start New Game button allows players restart a new game, Load Chapter allows players to view the levels, Setting (Figure 2.8) is to adjust the brightness of the game and view the game credits and game controls, while Exit button is to allows players quit the game.

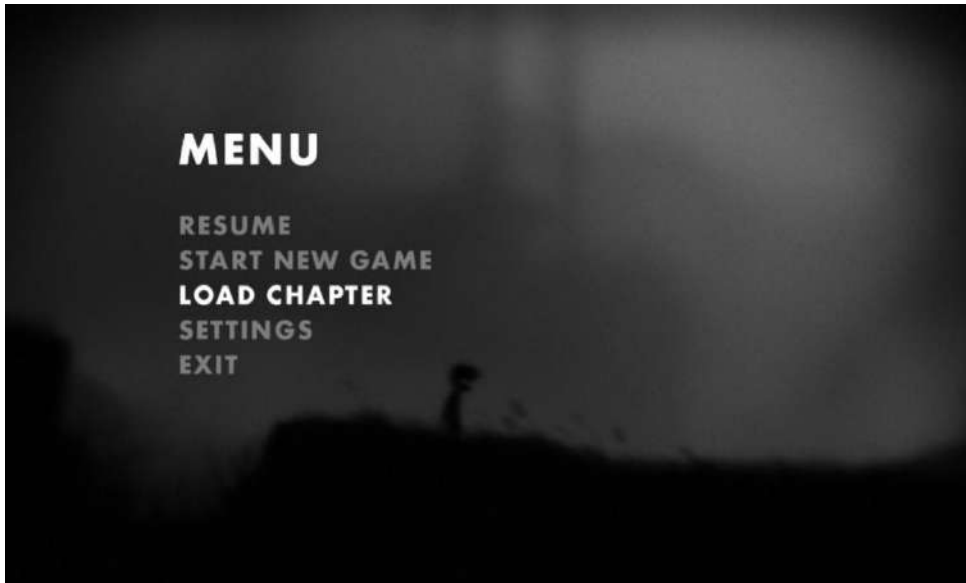


Figure 2.7 The Main Menu UI of LIMBO



Figure 2.8 The Setting UI of LIMBO

Figure 2.9 is the game UI of LIMBO. Presented in black and white, the game uses lighting, film grain effects, and minimal ambient sound to create the eerie atmosphere usually associated with the horror genre. Players are required to control the character to move and across the obstacles until they reach the end.



Figure 2.9 Game UI of LIMBO

2.2.3 The Final Station



Figure 2.10 The Final Station

The Final Station is a single-player side-scrolling shooter with a train simulator and exploration elements. Developed by Do My Best Games and published by tinyBuild, it was released on August 30, 2016 for Windows, MacOS, Linux and Xbox One, and on September 2, 2016 for PlayStation 4.

The Final Station includes two types of gameplay. First is the **Train Interior** part of the game, where players must balance caring for the survivors they rescue, feeding or healing them, and taking care of the train by completing small mini-games to keep the experimental train running until they reach the next train station. The second part of the

game is the **In-game Battle** that takes place at most train stations. In the game, players will explore an apocalyptic side-scrolling level that requires strategic planning to complete. There are different types of enemies to make the required level and strategy more difficult. Players need to find hidden codes at each station to get the train moving again, and find resources like ammo, food, and survivors.

Based on Table 2.3, the developer of The Final Station is Do My Best Games. The target player of the game is a teenager 13 years old or older. The game engine used to develop the game is Unity. This game is available on Windows, MacOS, Linux, Xbox One, PlayStation 4, Nintendo Switch. The Final Station is a single-player puzzle-adventure platformer. The minimum system requirements to run this game are operating system (OS) that is Windows XP or later, processor of 1GHz, minimum 1GB RAM of memory, graphics is Built in toaster, and a 2GB available storage space.

Table 2.3 Information and Minimum Requirements of The Final Station

TYPE	CONTENT
Developer	Do My Best Games
Target Player	Teenager 13 years old or older
Game Engine	Unity
Platform	Windows, MacOS, Linux, Xbox One, PlayStation 4, Nintendo Switch.
Genre	Side-scrolling shooter, train simulator, adventure
Mode	Single Player
Operating System	Windows XP
Processor	1GHz
Memory	1GB RAM
Graphics	Built in toaster
Storage	2 GB available space

Figure 2.11 is the Main Menu of The Final Station. There are 3 buttons provided which are Start Game, Setting and Quit. The Start Game button allows players to play the game, Setting button (Figure 2.12) is to adjust the resolution, toggle of Fullscreen, language, sound and controls, while the Quit button allows players to quit the game.



Figure 2.11 The Main Menu UI of The Final Station

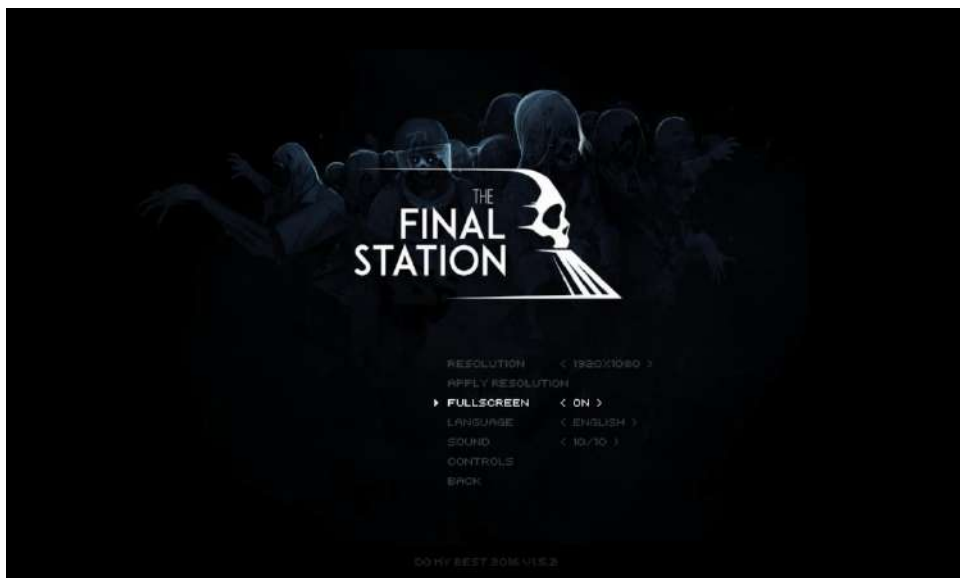


Figure 2.12 The Setting UI of The Final Station

Figure 2.13 is the game User Interface of The Final Station. In the game, players are required to control the main character to survive from zombie attacks, save the victims, and find notes, dialogue and resources like food, bullets etc (Figure 2.14) . Based on Figure 2.13, at the bottom of the interface will show the number of treatment boxes left, the red bar represents the health of the main character, the number of bullets left with the bullet fill in the weapon, number of food and key.



Figure 2.13 Game UI of The Final Station

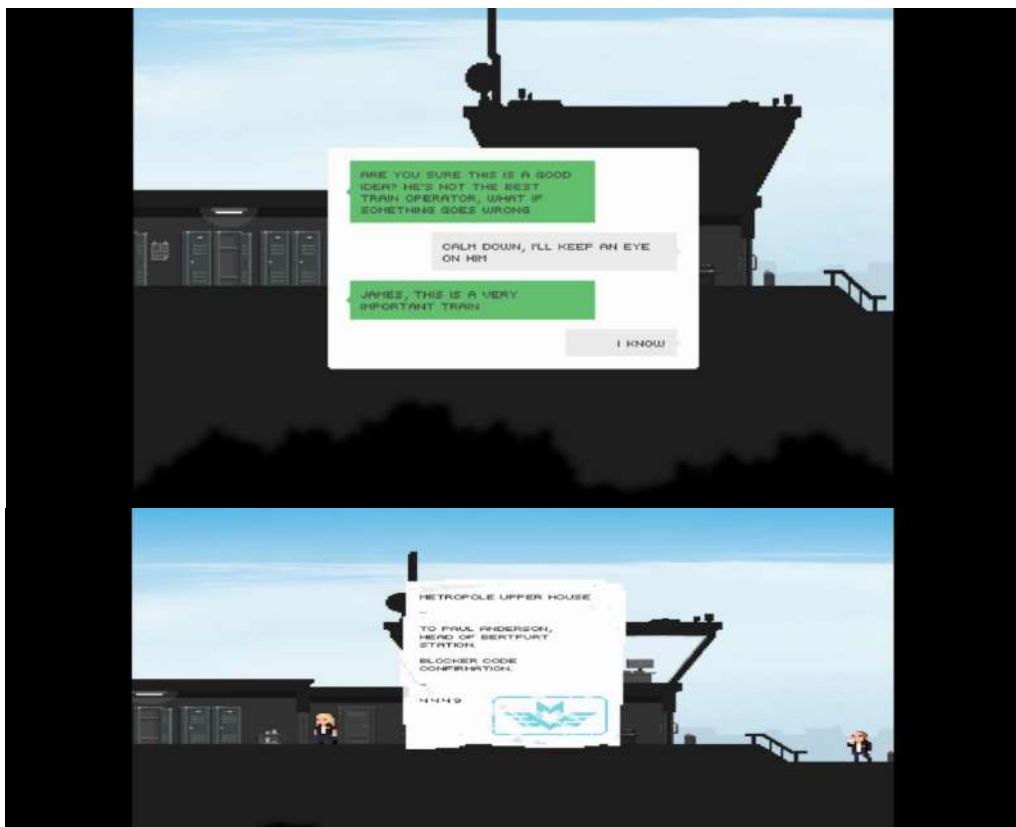


Figure 2.14 The Dialogue and Notes in The Final Station

Figure 2.15 is the User Interface of Shop in The Final Station. Players are allowed to buy and sell items in the shop. The amount of money owned will show at the top left of the dialogue.



Figure 2.15 The Shop UI of The Final Station



Figure 2.16 The Battery UI in Train Interior part of The Final Station

Figure 2.16, Figure 2.17, Figure 2.18 and Figure 2.19 are the User Interfaces in Train Interior part of The Final Station. Figure 2.16 is the battery UI which is a mini game where players need to balance the battery voltage, if players fail to do so, the train will lose electricity and stop functioning. Figure 2.17 is the message UI where NPCs will contact the players and players able to get the latest information or situation. Figure 2.18 is map UI where players can view the stations of the game. Figure 2.19 is the reserve UI where players are able to create the bullets and medical using the resources that they collect from the game.



Figure 2.17 The Message UI in Train Interior part of The Final Station



Figure 2.18 The Map UI in Train Interior part of The Final StationS



Figure 2.19 The Reserve UI in Train Interior part of The Final Station

2.3 Comparison Table of the Existing Game Applications

Table 2.4 is the comparison table of the existing game applications which are Toby: The Secret Mine, LIMBO, and The Final Station. Based on the table, the features and game design of Toby: The Secret Mine are like the LIMBO.

The features of Toby: The Secret Mine are that players are required to look for villagers that have been kidnapped, interesting puzzles designed for players to solve, and the game perform challenging missions in environments like the jungle, beautiful place art, like a Limbo setting. The advantages are surprising puzzle variety, great puzzle balance, a real beauty, charming design and a few gameplay surprises. Besides, the disadvantages of the game are the game is too short and a little replay with alternate endings, some control mechanics are weak and the colour of the character is the same with the game objects, making players have visibility difficulty in several levels.

The features of LIMBO are that the game provide a deep environments with lots of dangers for players to enjoy, provide great storytelling at each Chapter, no loading system applied in game for players to play the game smoothly and provide wonderful puzzles with special properties. The advantages of this game are the amazing art and visual design that help players easily integrate into the game, the game provides several interesting organic puzzles that do not get boring, and a captivating visual style. The

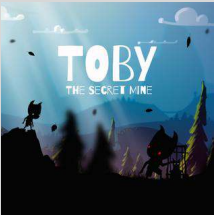
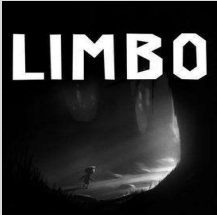

disadvantages are the game could have been longer and more levels, the late-game platforming makes players feel very slightly clunky, vague achievements, and involves a fair amount of trial and error.

Next, the features of The Final Station are that the game provides a story that goes through 5 chapters, each ending with a large, inhabited station, the infected are ready to stop the player every time they get off the train in search of supplies, players need to keep the train running and craft valuable items from the collected resources, and players have to alert of the limitation of the supplies and ammunition. The advantages are a great mix of fast-paced action and strategy, amazing eerie style, and solid mishmash of mechanics. The disadvantages are predictable mission structure, and the game controls take a little getting used to.

Table 2.4 Comparison of the Existing Game Applications

Game Application Name	Toby : The Secret Mine 	LIMBO 	The Final Station 
Developer	Lukáš Navrátil	Playdead	Do My Best Games
Target Player	Everyone 10 years old or older	Teenager 18 years old or older	Teenager 13 years old or older
Game Engine	Unity	Box2D	Unity
Platform	Windows, Linux, OS X, iOS, Android, Xbox One, Wii U, PlayStation 4, Switch	Xbox 360, PlayStation 3, Windows, OS X, Linux, Xbox One, PlayStation 4, PlayStation Vita, Nintendo Switch, iOS, Android	Windows, MacOS, Linux, Xbox One, PlayStation 4, Nintendo Switch.
Genre	Puzzle, Adventure, Platform	Puzzle, Adventure, Platform	Side-scrolling shooter, train simulator, adventure
Mode	Single Player	Single Player	Single Player
Operating System	Win 7 or later	Windows XP, Vista, 7	Windows XP
Processor	Intel Core i3 or better	2GHz	1GHz

Table 2.4 Continued

Game Application Name	Toby : The Secret Mine 	LIMBO 	The Final Station 
Memory	2GB RAM	512 MB	1GB RAM
Graphics	Intel HD 4000 or better	Video Card 5 years or younger, Shader Model 3.0	Built in toaster
DirectX	version 9.0	version 9.0c	-
Storage	800 MB available space	150 MB available space	2 GB available space
Features	<p>Looking for villagers that have been kidnapped. Interesting puzzles to solve. Perform challenging missions in environments like the jungle. Beautiful place, like a Limbo setting.</p>	<p>Deep environments with lots of dangers. Great storytelling at each Chapter. No loading system. Wonderful puzzles with special properties.</p>	<p>A story that goes through 5 chapters, each ending with a large, inhabited station. The infected are ready to stop the player every time they get off the train in search of supplies. Keep the train running and craft valuable items from the collected resources. Supplies and ammunition are very limited.</p>
Advantages	<p>Surprising puzzle variety. Great puzzle balance. A real beauty. Charming design. A few gameplay surprises.</p>	<p>The art and visual design integrated into the game. Organic puzzles that do not get bored. Captivating visual style.</p>	<p>Great mix of fast-paced action and strategy. Great eerie style. Solid mishmash of mechanics.</p>
Disadvantages	<p>Short, Little replay versus alternate ending. Some control mechanics are weak. In some levels, the colour of the character is the same with the game objects, making players have visibility difficulty.</p>	<p>Could have been longer. Late-game platforming can feel very slightly clunky. Vague achievements. Involves a fair amount of trial and error.</p>	<p>Predictable mission structure. Controls take a little getting used to.</p>

2.4 Chapter Summary

In conclusion, there are several elements to pay attention to, such as game design, gameplay, and minimum system requirements. A perfect game design grabs the player's attention, increases their interest in the game, and helps players integrate into the game more easily. Also, the gameplay should be logical and easy to understand. The gameplay affects the player's gaming experience. If the gameplay is too complicated, the player will spend more time getting used to the controller and game rules. Also, the minimum system requirements are important in game development as it affects the settings of the game application.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Methodology refers to the overall strategy and rationale of the project. It involves studying the methods used in the field and the theories or principles behind them in order to develop methods that match the goals.

The Software Development Life Cycle (SDLC) is a framework used by development teams to produce high-quality software in a systematic and cost-effective manner. Both large and small software organisations use the SDLC methodology. The software development life cycle (SDLC) provides organisations with a systematic, step-by-step approach to developing successful software by gathering initial requirements for a new product. There are several SDLC models that can be used for 2D game development such as Waterfall, Agile, ADDIE, RAD, Spiral etc.

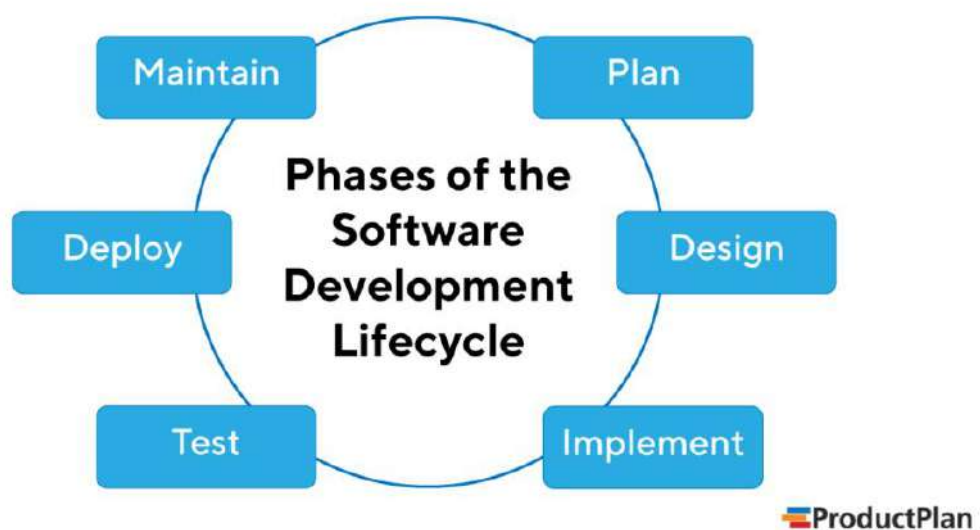


Figure 3.1 Phase of the Software Development Lifecycle (SDLC)

3.2 Spiral Model

The Spiral Model is an iterative development method. Iteration means the steps are performed repeatedly. This means the project is conceived, designed, prototyped, tested, and the test results are put into the next iteration of the same steps. The Spiral Model is thus an evolutionary process.

The spiral model for game development is a simple and elegant technique that is often used to develop games efficiently. Inexperienced game developers tend to design a game with a lot of features and then start implementing all of them without any prioritisation. It is always a good practice to define which features should be developed first and which should be developed later.

Figure 3.2 is a simple diagram of the Spiral Model. Based on the Figure 3.2, in game development, there are 4 typical phases in the Spiral Lifecycle Model which are Design and Planning, Implement, Play Test, and Evaluate.

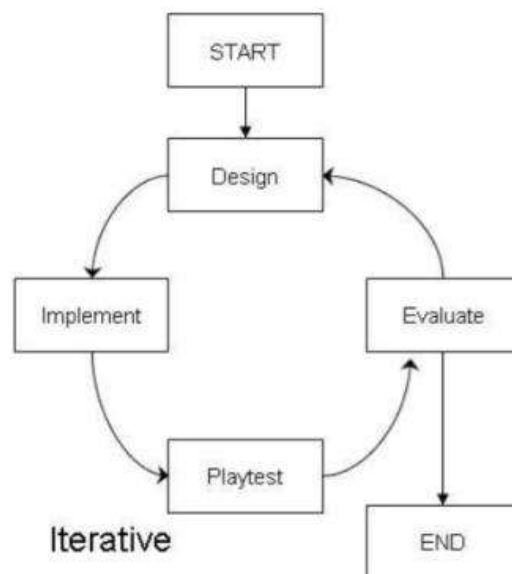


Figure 3.2 Simple Diagram of Spiral Model

3.2.1 Planning & Design Phase

At this phase, developers begin to analyse and plan the game ideas. Developers will plan the outline and storyboard of the game. Planning is followed by designing the

game mechanics, characters, gameplay, level design, etc. This phase is important for developers who have guidelines or directions to develop the game.

3.2.2 Implement Phase

At this phase, after planning and design are complete, the developer will begin the game programming process. Start coding from here, developing the features that have been planned and designed in the previous stage. This phase can take a long time to develop a working prototype and ensure that the prototype is as expected.

3.2.3 Play Test Phase

During this phase, developers will test the game by selecting a group of users who have played the prototype completed during the implementation phase to identify flaws in gameplay, level design, and other essential elements, and to find and fix bugs and glitches. This happens at multiple stages of the video game design and development process. This is a quality control method that occurs at multiple stages of the video game design and development process.

3.2.4 Evaluate Phase

At this phase, developers need to understand what is right and wrong and have a new perspective, and then iterate back to the planning and design stage again.

3.3 Spiral Model in the Development Process of the Project

Spiral models are an excellent choice for large and complex projects. The incremental nature of the model allows developers to break down a large project into smaller pieces, working on one feature at a time, ensuring nothing is missed. Also, since prototyping is done incrementally, it is sometimes easier to calculate the cost of the entire project.

Figure 3.3 is a detailed diagram of the Spiral Model. Based on Figure 3.3, the game development begins with determining the objectives in the Planning and Design Phase. After that, identify and resolve risks in Implement Phase, then development and test in Play Test Phase and finally plan the next iteration in Evaluate Phase.

Initially, requirements concepts, operational concepts and requirements plans are studied and planned during the planning and design phase, followed by the creation of the first prototypes during the implementation phase. Next, the first prototype will be tested, verified and validated as required by the planning and design phase plans. After that, start evaluating the results of the first prototype tests and plan the development plan during the evaluation phase. Then, defining the requirement of the second prototype, implement and test the prototype. Existing prototypes are evaluated in the same way as the previous prototype, and if necessary, another prototype can be developed according to the quadruple procedure described above.

The above steps are repeated until players are satisfied with the final result represented by the refined prototype and build the final system based on the refined prototype. The final game system is fully evaluated and tested. Routine maintenance is being carried out to prevent large-scale failures and minimise downtime.

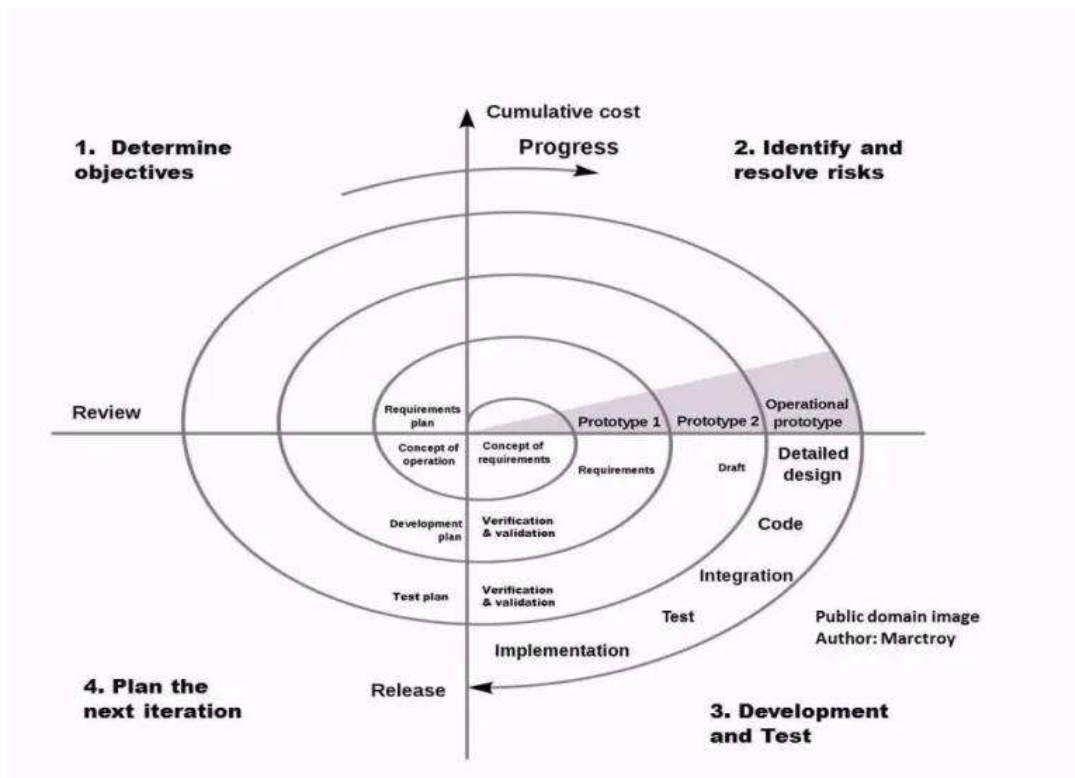


Figure 3.3 Detailed Diagram of Spiral Model

Table 3.1 Task Distribution for Spiral Model in the Project

Phase	Description
Planning & Design	<ul style="list-style-type: none"> - Brainstorming ideas - Study on theme of the project and identify the background issues of the project - Identify the problem statements, objectives, and scopes - Study and analysis on the existing game application - Do survey on user requirements and system requirements - Study and research the software development lifecycle (SDLC) - Identify and define the methodology used in the project - Decide the hardware and software to be used to develop the project - Plan the game content, game mechanic and game technology - Design and sketch the game assets and animation - Create Gantt chart, use case diagram and context diagram - Sketch the flow chart - Storyboarding <p>For the second cycle or further cycles of the spiral model</p> <ul style="list-style-type: none"> - Identify and study the solution for the problems stated in the feedback - Identify the changes or improvements
Implement	<ul style="list-style-type: none"> - Create the game assets - Coding and scripting - Create a playable prototype <p>Last cycle</p> <ul style="list-style-type: none"> - Create final complete game programming
Play Test	<ul style="list-style-type: none"> - Play and test the prototype - Collect the UAT and User feedback test from a group of players that test the prototype <p>Last cycle</p> <ul style="list-style-type: none"> - Final test of the game and implementation
Evaluate	<ul style="list-style-type: none"> - Analysis the feedbacks - Determine and list out the correct parts and wrong parts - Improve or maintain the correct part - Plan the solution for the wrong part <p>Last cycle</p> <ul style="list-style-type: none"> - Release the game

3.4 Project Requirements

Project requirements are the features, functions, and tasks that a project needs to complete in order to be considered successful or at least completed. It provides everyone involved with a clear set of parameters to work towards and identify the various requirements to be achieved by stakeholders. The project requirements for this

project will be identified as Functional and Non-Functional Requirements, Constraints and Limitations.

3.4.1 Functional Requirements

- The game should allow players to click the New Game, Continue, Highscore and Quit buttons in the Main Menu of the game.
- The game should allow players to view the score, new high score and high score when click the Highscore button on the Main Menu.
- The game should allow players to move the character left, right, up and down by pressing the arrow key or WASD on the keyboard.
- The game should allow players to control the character to open furniture, talk to NPCs and read note by pressing space on the keyboard.
- The game should allow players to control the character to open the door by pressing E on the keyboard.
- The game should allow players to shoot by pressing Q on the keyboard.
- When applying the AI function on enemies, enemies should be able to move and attack players automatically.
- The game should be able to display the score on the top left of the game screen.
- The game should be able to show the life of the character and enemies.
- The game should allow players to click the Pause button on the right top of the game screen to pause, continue, replay and quit the game.
- The game should be able to display the animation, background music and sound effect.
- The game should be able to display the score and high score after players complete the game.

3.4.2 Non-Functional Requirements

Framerate

- The minimum framerate of the game must be 20 frames per second, and the average framerate must be greater than 30.
- The framerate of the game should be monitorable directly from the graphics engine.

Usability

- The high score must be able to support 1 million concurrent user records.

Response Time

- The button response time must be less than 0.5 seconds after being clicked and the maximum response time is 2 seconds.
- The response time of scene navigation must be less than 5 seconds and the maximum response time is 10 seconds.
- The response time of the high score must be less than 2 seconds and the maximum response time is 5 seconds.

Required Resources

- The game must be able to run with at least 1024 MB of RAM.
- The game must use less than 1GB of hard drive space.

Platform

- The game must be able run in Window XP or later.

3.4.3 Constraints and Limitation

- There is no level map provided. Players are able to continue the game at the latest checkpoint reached.
- The project is a single player game which is designed to be played by one player only.
- The game platform is only for PC.
- Players need to analyse the notes or dialogues in the game to predict the truth.

3.5 User Requirements

A questionnaire form with 15 questions has been done through Google Form. There are about 15 respondents who have answered the Google Form. The questionnaire is divided into 2 sections which are Demographic Profile and “What You Think of The Kidnapping 2D Game?”. The first section is to collect the respondent's basic data, while the second section is to collect the knowledge and the understanding about the kidnapping game.

Table 3.2 shows the Data Responses From Google Form. Summary is most of the respondents have played Adventure, Puzzle and Platform 2D games before through PC. Most respondents know what Kidnapping is but only a few respondents did read or search the articles about Kidnapping, and only two respondents played kidnapping 2D games before. Most respondents strongly agree that the 2D games can raise public awareness of kidnapping, reduce the risk of being kidnapped, and teach players how to avoid being kidnapped. They also agree that 2D game should be easy to control and the instruction should be easy to understand. All the respondents respond that they will recommend the game to their friend.

Table 3.2 Data Responses from Google Form

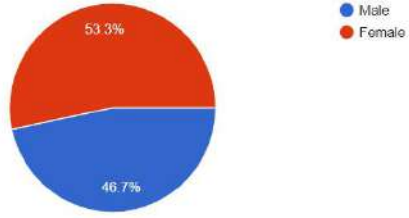
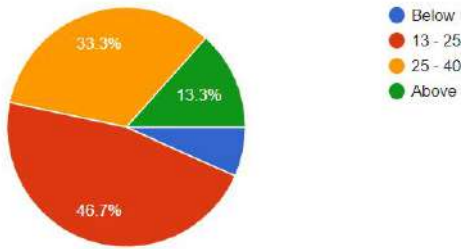
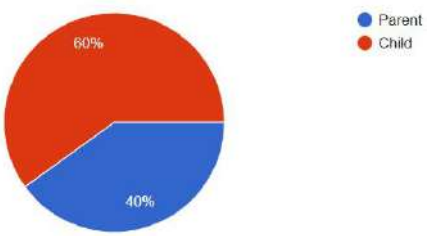
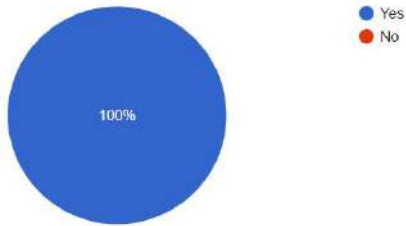
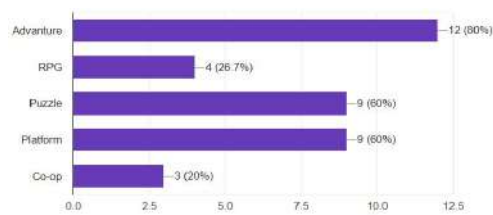
No	Question	Result
Section 1 : Demographic Profile		
1	Gender : Male / Female	<p>Result :</p> <p>In the total of 15 respondents, 8 (53.3%) respondents are female, and 7 (46.7%) respondents are male.</p> 
2	Age : below 13 13 - 25 25 - 40 above 40	<p>Result :</p> <p>In the total of 15 respondents, 1 (6.7%) respondent is below 13 years old, 7 (46.7%) respondents are 13-25 years old, 5 (33.3%) respondents are 25-40 years old, and 2 (13.3%) respondents are above 40 years old.</p> 
3	Role in family : Parent / Child	<p>Result :</p> <p>In the total of 15 respondents, 9 (60%) respondents are children in a family, and 6 (40%) respondents are parents in a family.</p> 
4	Do you play 2D games before? : Yes / No	<p>Result :</p> <p>In the total of 15 respondents, all the 15 (100%) respondents have played 2D games before.</p> 
5	What type of 2D games do you play? : Adventure / RPG / Puzzle / Platform / Co-op	<p>Result :</p> <p>In the total of 15 respondents, 12 (80%) respondents played adventure games, 4 (26.7%) respondents played RPG games, 9 (60%) respondents played puzzle and platform games, and 3 (20%) respondents played co-op games.</p> 

Table 3.2 Continued

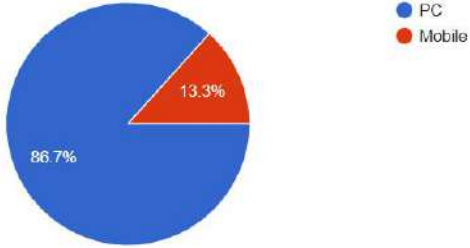
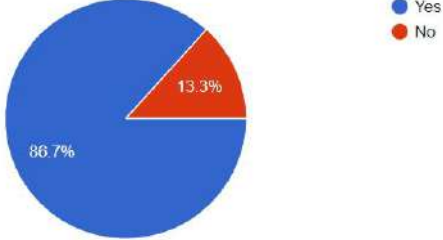
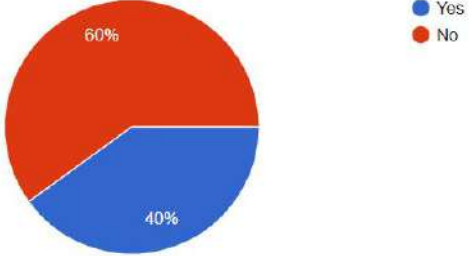
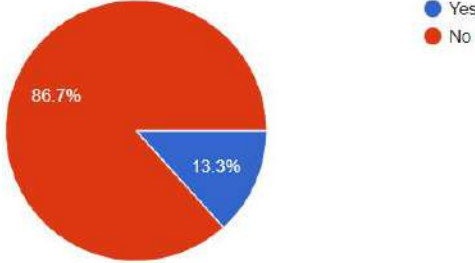
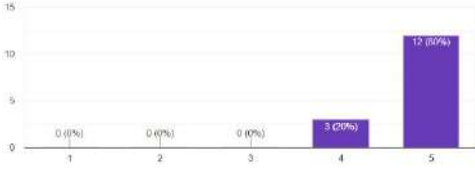
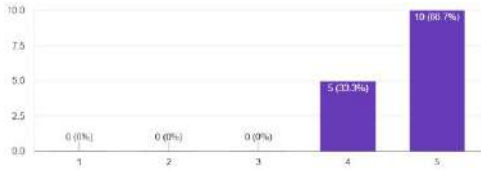
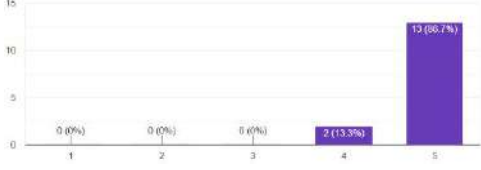
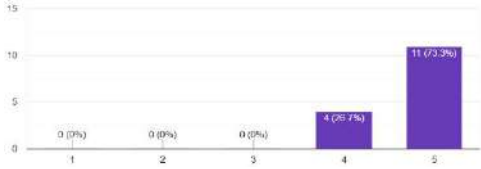
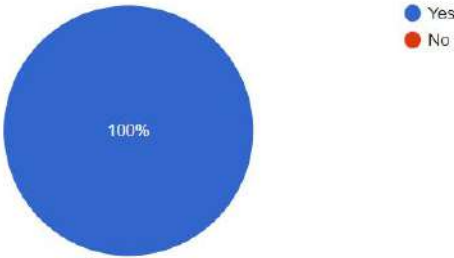
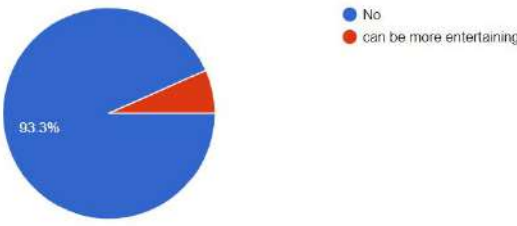
No	Question	Result
6	What device do you use most often to play games? : PC / Mobile	 <p>Result : In the total of 15 respondents, 13 (86.7%) respondents use PC, and 2 (13.3%) respondents use mobile.</p>
Section 2 : What You Think Of The Kidnapping 2D Game?		
7	Do you know what kidnapping is? : Yes / No	 <p>Result : In the total of 15 respondents, 13 (86.7%) respondents know what the kidnapping is, and 2 (13.3%) respondents do not know.</p>
8	Do you read / search the articles about kidnapping ? (News / Website / Magazine etc) : Yes / No	 <p>Result : In the total of 15 respondents, 9 (60%) respondents did not, and 6 (40%) respondents read and searched the articles about kidnapping.</p>
9	Do you play kidnapping 2D games before? : Yes / No	 <p>Result : In the total of 15 respondents, 13 (86.7%) respondents did not play kidnapping 2D games, and 2 (13.3%) respondents did.</p>
10	2D games can raise public awareness of kidnappings.	 <p>Result : In the total of 15 respondents, 12 (80%) respondents strongly agree, and 3 (20%) respondents agree with the statement.</p>

Table 3.2 Continued

No	Question	Result
11	2D games can reduce the risk of being kidnapped.	 <p>Result : In the total of 15 respondents, 10 (66.7%) respondents strongly agree, and 5 (33.3%) respondents agree with the statement.</p>
12	2D games can teach players how to avoid being kidnapped.	 <p>Result : In the total of 15 respondents, 13 (86.7%) respondents strongly agree, and 2 (13.3%) respondents agree with the statement.</p>
13	The 2D games should easy to control and understand the instruction.	 <p>Result : In the total of 15 respondents, 11 (73.3%) respondents strongly agree, and 4 (26.7%) respondents agree with the statement.</p>
14	Would you recommend this kidnapping game to your friends? : Yes / No	 <p>Result : In the total of 15 respondents, all the 15 (100%) respondents will recommend the game to their friends.</p>
15	Do you have any suggestions to make the kidnapping game more successful?	 <p>Result : In the total of 15 respondents, 14 (93.3%) respondents have no suggestion , and 1 (6.7%) respondents suggested that the game can be more entertaining.</p>

3.6 Proposed Design

3.6.1 Flowchart

Figure 3.4 shows the flowchart of the Main Menu. From the start, the system will display the Main Menu with 3 buttons which are New Game, Continue and Quit. If the New Game button is clicked, then begin and display the Level 1, else if the Continue button is clicked, then the system will back saved level, else if the Quit button is clicked, then system will quit and end the flowchart, else back to display the Main Menu and the first decision.

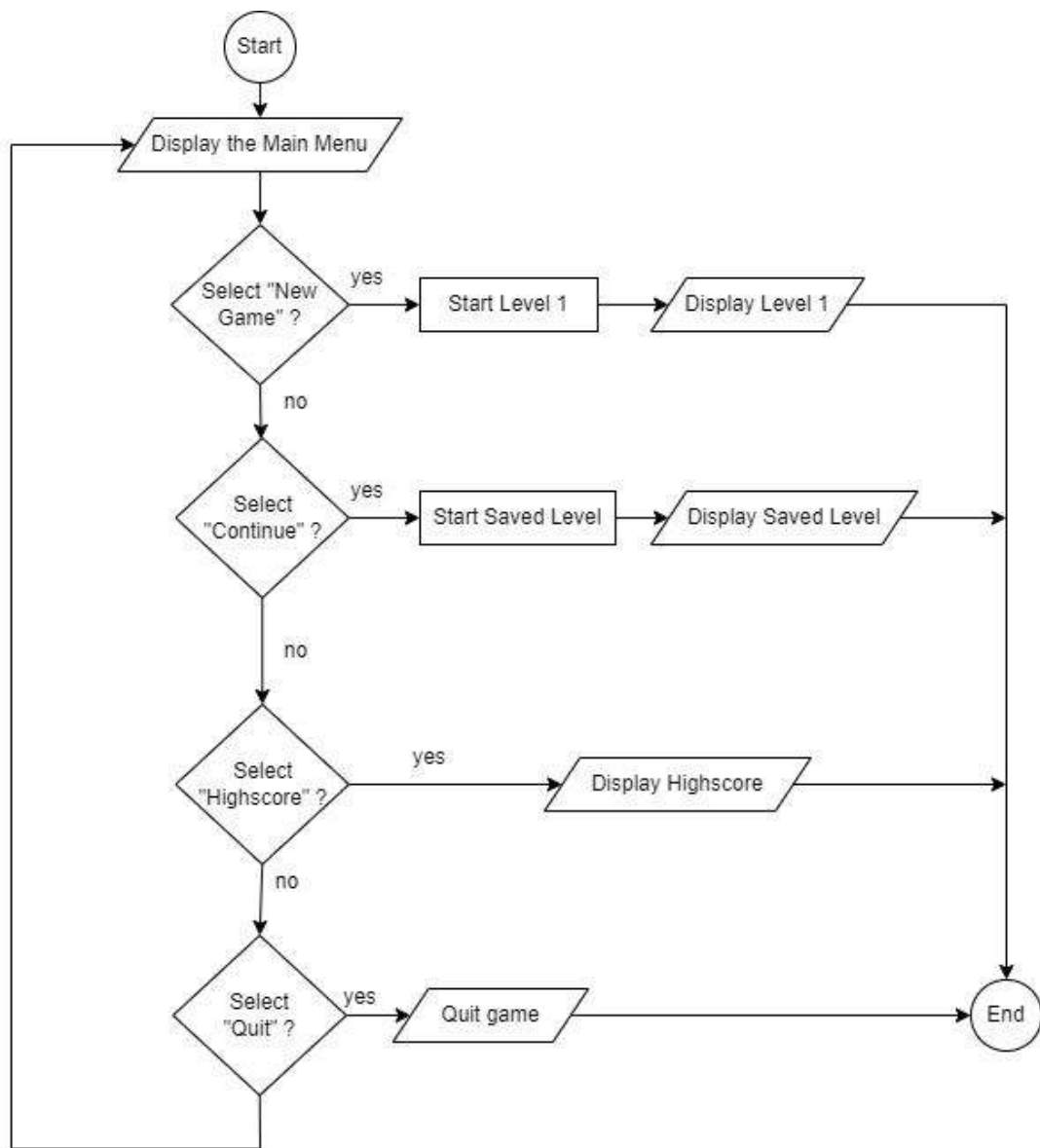


Figure 3.4 Flowchart of the Main Menu

Figure 3.5 shows the flowchart of the Game Play in a Level. From the first, if the player chooses a New Game in the Main Menu, then the system will display the Level 1, the PlayerHealth will be 100 and the score will be 0, else the PlayerHealth and the score will be currentHealth and currentScore, which brings from previous level. If the player has not finished the task of the level, the system will go to the loop in which the player moves, meet, and kill the enemy, find and collect the key to open the door. Once the player finishes the task, then the player will move to the next level and end the flowchart.

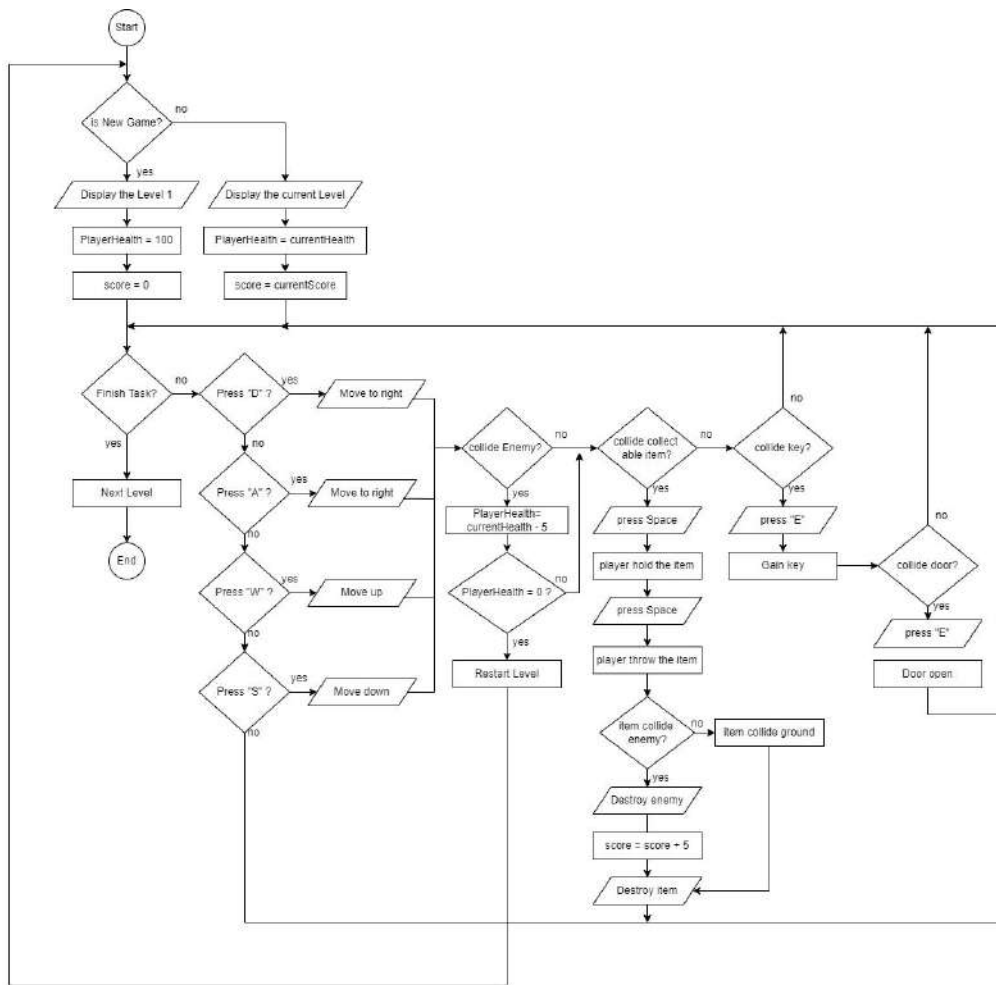


Figure 3.5 Flowchart of the Game Play in a Level

Figure 3.6 shows the flowchart of the Pause Menu. In the game, if the player presses the Pause button, then the Pause Menu will display. If the player presses Continue, then the player will go back to the game, else if the player presses Replay, then the player will replay the level, else if the player presses Quit, then the player will go back to the Main Menu.

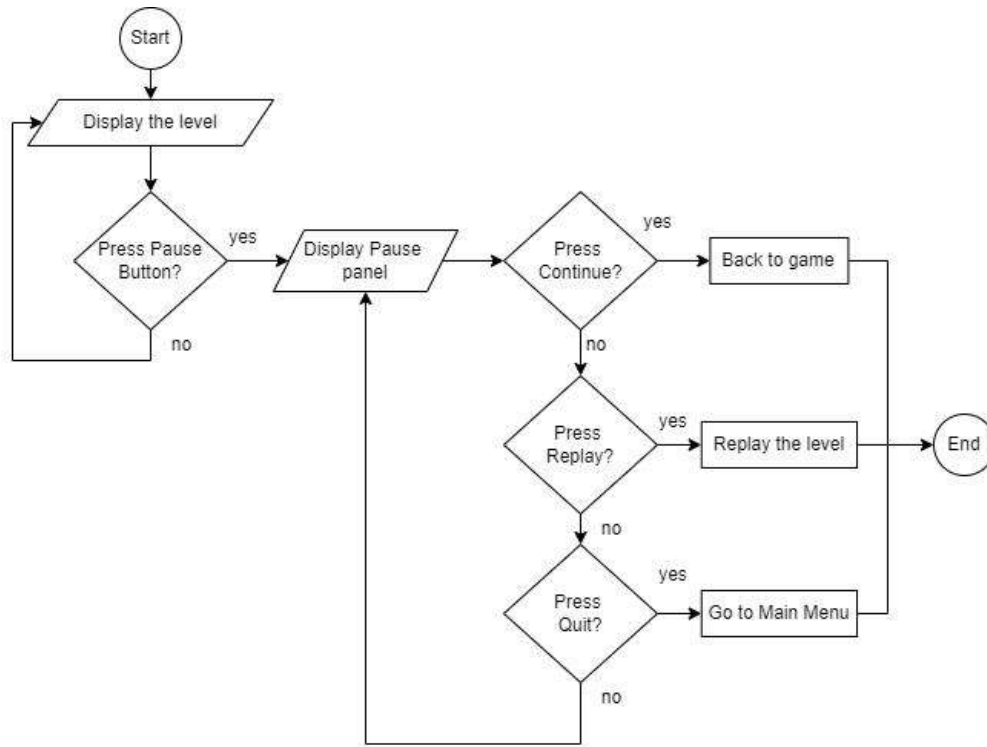


Figure 3.6 Flowchart of the Pause Menu

3.6.2 Context Diagram

Figure 3.7 is a context diagram between a player with the system. The system will receive the input data such as a pressed button from the player, then display the expected output to the player.

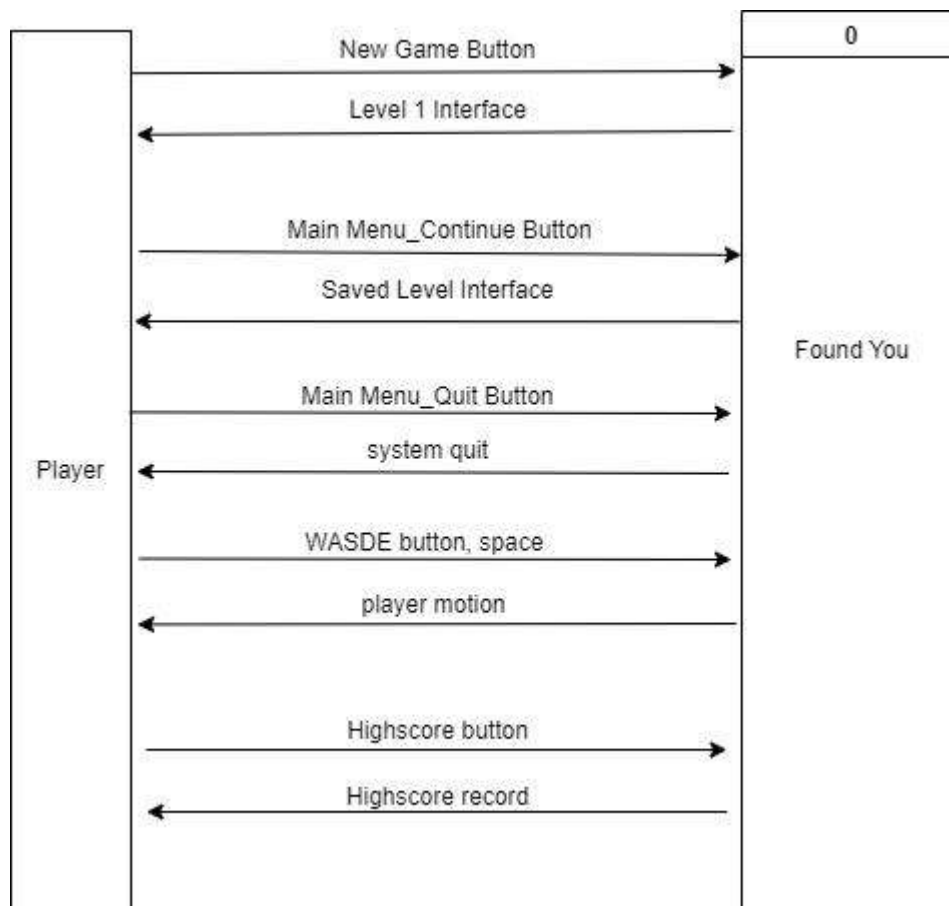


Figure 3.7 Context Diagram of the Project

3.6.3 Use Case Diagram

Figure 3.8 is a use case diagram that shows what a player is able to do in the game system. In the game, the player should be able to view the instructions of the game, view the game story, control the character, click buttons, view the high score and insert names for the high score.



Figure 3.8 Use Case Diagram of the Project

3.6.4 Activity Diagram

Figure 3.9 is an activity diagram of the project. From the start, the system will display the Main Menu for players to select. If Quit is selected, the system will quit the game, else New Game or Continue is selected, the system will display the level interface and game play instruction. Once the player controls the character, the system should display the correct animation. Next, the system shows the health bar where the health bar is empty, the system will display the lose panel, else the system will show the score of the interaction made by the player. If the procedure is not finished, the system will wait until the player controls the character and the procedure is finished. Next, the system will show the win panel, the game score and the high score. Players can view the high score.

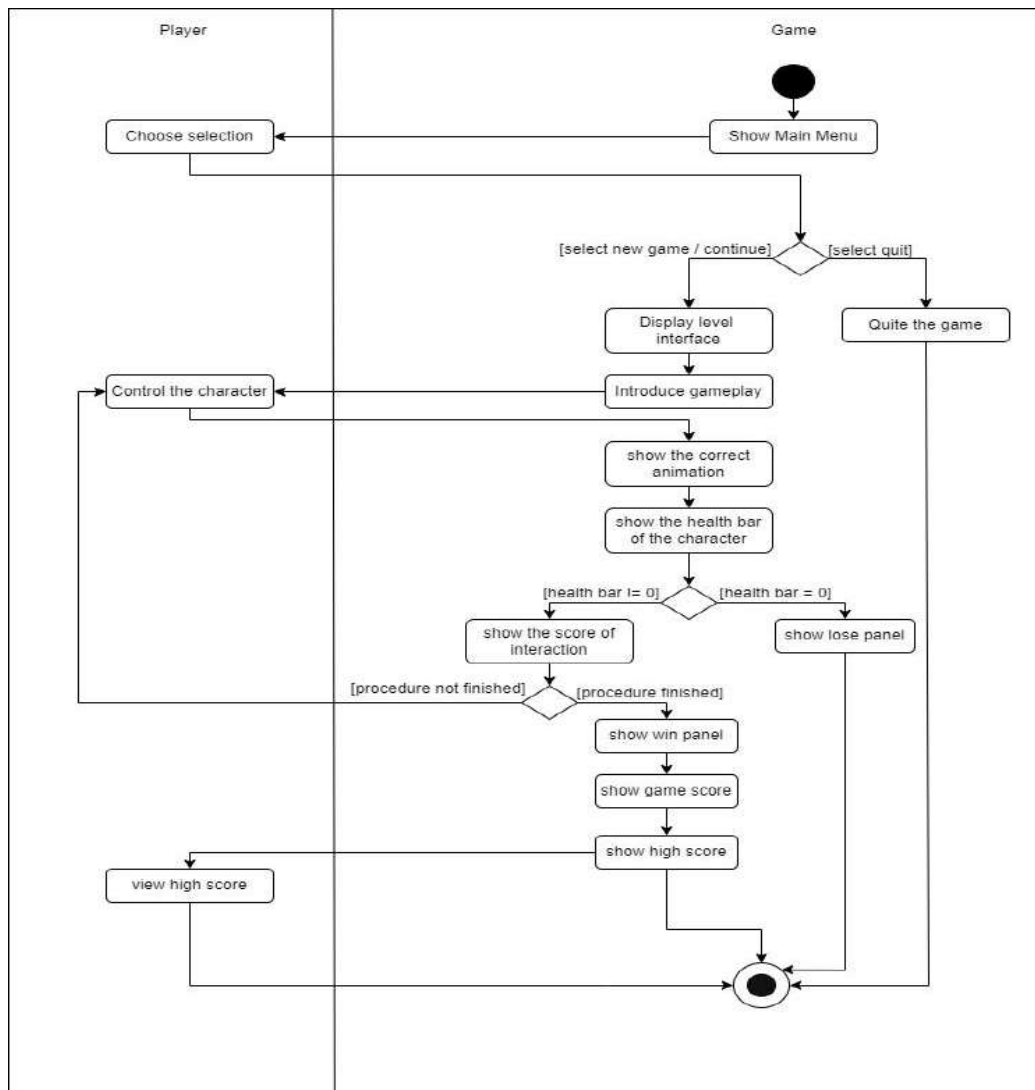


Figure 3.9 Activity Diagram of the Project

3.6.5 Game Content

- Game Genre

The game is an adventure puzzle game, players need to explore and complete tasks, collect clues for story prediction.

- Environment

The environment of the game is a small town. Level 1 is Anne's (Main Character) House, Level 2 is Mini Mart, Level 3 is Emily's House, Level 4 is School.

- Characters

Anne (Main Character)

- Emily's best friend
- Controlled by players to save Emily.

Emily (NPC)

- Anne's best friend
- kidnapped in a abandoned house

Residents (NPC)

- NPC in the game
- Their role is to give task and clues

Enemy (NPC)

- Give damage of 5 to character
- able to be killed.

- Game Objects

Key, Door, Note, Bandage, Cabinet, Iron Cabinet, Book Rack, Table3.

- Narrative

The game begin in Anne's house. Anne's best friend, Emily comes to Anne's room and ask to play hide and seek, where she hide and Anne try to find her. Then, Emily goes to

hide, after that Anne comes out from her room and starts to find clues in all levels to find Emily.

In Level 1 is in Anne's House. In the first there are 4 tutorials which are Player Movement tutorial, Open door tutorial, Search Furniture tutorial and Read note tutorial. After that Anne will found first clue leave by Emily which is clue 1 and a poster about "Do Not Open Doors For A Stranger". Then Anne will found a key to open the locked door for the Anne's House. When Anne approach the door, a stranger will ask Anne to open the door by lying that Anne's mother ask him to take care of her. Anne will be kidnapped if Anne open the door, else if Anne do not open the door, the kidnapper will walk away from Anne's house and Anne able exit the house. On the way, Anne will meet a resident and the resident will tell Anne where Emily goes. In front there is a small building, and in the building Anne will found a Clue 2 and a Map from Emily.

In level 2, there is a small building and Anne will found another clue which is Clue 3 from Emily. After that Anne will reach Mini Mart, and outside the Mini Mart there is a resident (child) sitting on a bench and when talk with him, the resident will tell Anne that his mother told him do not go to the back alley because it is dangerous. In the Mini Mart, a resident tell Anne that he saw a stranger walking around in the back alley and the resident offer a help if Anne harassed by the stranger. For this level, Player will found 2 message to avoid kidnapper which are Poster about "Avoid Less Crowded Places" and a Picture about "Run and Get Help if kidnapper Approach". When Anne enter the Back Alley Anne will meet the kidnapper. The kidnapper start harassing Anne, then Anne has to run away but the kidnapper will chase from behind. Anne back to the Mini Mart and ask help from the resident.

In Level 3, Anne come to Emily's House. On the way to Emily's House, Anne found Emily's Shoes on the floor. Anne may predict that Emily is in trouble. In Emily's House, Anne found 2 messages which are a Picture about "Do Not Ride Stranger's Car" and a Picture about "Do Not Take Anything From A Stranger". After Anne exit Emily's House, Anne will meet the kidnapper and the kidnapper will offer a ride to Anne. Anne have to reject the offer from the kidnapper else Anne will be kidnapped.

In Level 4, Anne comes to Playground and did not saw Emily. Anne meet a Police and the police tell Anne that they get report about the kidnapper is around the School, thus

want Anne to be careful and tell the police if know anything about the kidnapping. In the School, Anne found 2 messages which are a Picture about “Kick or Bite the Kidnapper to Escape” and a Picture about “Run and Report to Police if know about a Kidnapping”. After that Anne will accidentally heard the conversation between kidnappers and found that the victims are hide in Abandoned House. At the same time, kidnappers saw Anne and try to catch her. Anne run away and meet with the police. Anne give the correct information to the police and police successfully arrest the kidnappers at the Abandoned House.

3.6.6 Game Mechanic

- Game Rules

Players are required to press WASD to move the character up, left, down and right. Press Space is to talk, read, hold and throw items, while press E to collect keys and open doors. Players have to move the character to find the clues (notes) and explore all the game places until they reach the end. Players are given a health bar, if the health bar becomes 0 then the players will respawn (there will provide different respawn points), Players can gain health by finding the bandage. There also have some enemies (monsters, wild animals etc.) that will harm the character, trying to avoid them.

- Rewards

The rewards are keys to open doors, bandage to recover the health bar, scores gained from killing enemies and the victory scene.

- Punishments

The punishments are that the character's health bar will be deducted by 5 if attacked by enemies, players are required to respawn if the character's health bar becomes empty, and the lose scene.

- Victory Condition

Players are required to avoid the health bar becoming empty and successfully explore all the levels.

- Challenges

There are enemies attacking the player. The enemies will sense the character, if the character gets into their view, they will start to chase the character.

3.6.7 Game Technology

- Network

This game is a single player game, no internet is needed.

- Graphics

2D pixel drawing and the design will be Malaysia style, the main colour is black, white and grey (dark colour) to create a mysterious atmosphere. Smooth animation and narrative will provide.

- Audio

The background music will use the soft and slow BGM and sound effect will provide an increased gaming experience.

3.6.8 Storyboard

Table 3.3 Storyboard of the Game

FRAME 01 (Main Menu)	
PROJECT TITLE: Found You	SCENE: Main Menu Scene



DESCRIPTION:

This is the Main Menu scene of the game application, players are able to view the scene once players open the game.

BUTTON:

New Game - Start a new game and navigate to Game Beginning Instruction scene
 Continue - Continue the previous saved level
 Highscore - View the score, new high score and high score

GRAPHIC:

Background Image

AUDIO:

Button Clicking Sound Effect

MUSIC:

Background Music - soft Music Box

ANIMATION:

None

PREVIOUS SCENE:

None

Table 3.3 Continued

FRAME 02 (High Score)	
PROJECT TITLE: Found You	SCENE: Main Menu Scene



DESCRIPTION:

This is the Highscore scene where allow players to view the score, new high score and high score. Player also able to back to Main Menu and Clear the high score data.

GRAPHIC:

Background Image

BUTTON:

Back - back to Main Menu

Clear - clear the high score data

AUDIO:

Button Clicking Sound Effect

MUSIC:

Background Music - soft Music Box

ANIMATION:

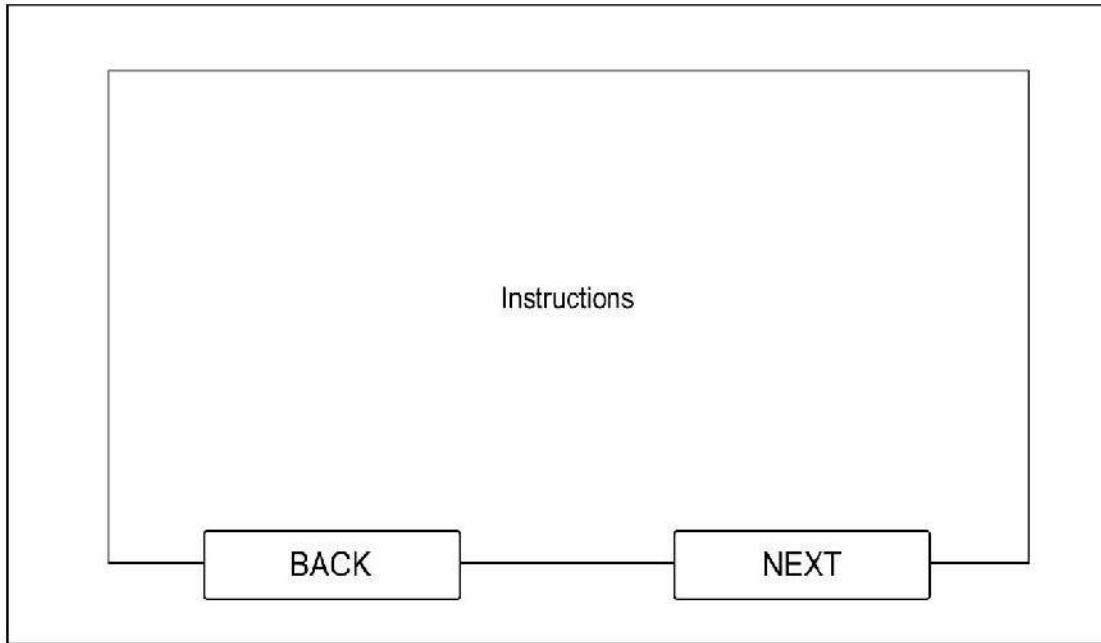
None

PREVIOUS SCENE:

None

Table 3.3 Continued

FRAME 03 (Game Beginning Instruction)	
PROJECT TITLE: Found You	SCENE: Game Beginning Instruction Scene



DESCRIPTION:

This is the Game Beginning Instruction Scene which allow players to understand the purpose of the game.

GRAPHIC:

Background Image

BUTTON:

Back - back to Main Menu

Next - navigate to level 1

AUDIO:

Button Clicking Sound Effect (paper flip sound)

MUSIC:

None

ANIMATION:

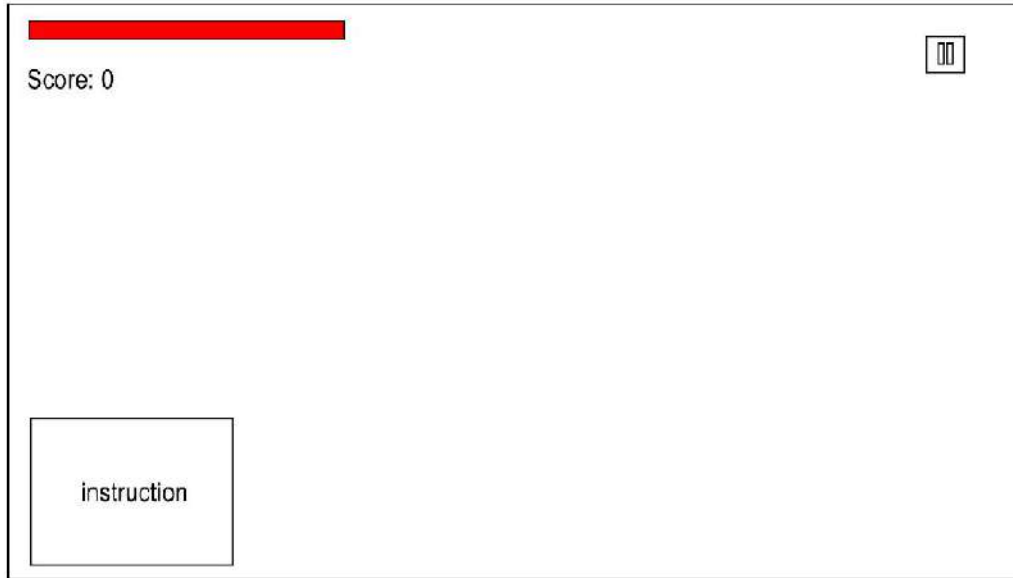
None

PREVIOUS SCENE:

None

Table 3.3 Continued

FRAME 04 (Game Player Interface)	
PROJECT TITLE: Found You	SCENE: Game Play Scene



DESCRIPTION:

This is the Game Play scene of the game, where the interface is consistent for all the level. The top left will display the player's health bar and score, bottom left display the player movement instruction, while top right display the Pause button.

GRAPHIC:

Background Image of level

BUTTON:

Pause - display the Pause Menu

AUDIO:

- Button Clicking Sound Effect (paper flip sound)
- Player walking and shooting Sound Effect
- Furniture opened Sound Effect
- Door open and locked Sound Effect
- Keypad button clicking, correct and wrong answer Sound Effect
- Enemy chase and attack Sound Effect
- Dialogue pop out Sound Effect
- Read Message Sound Effect

MUSIC:

Background Music for level

ANIMATION:

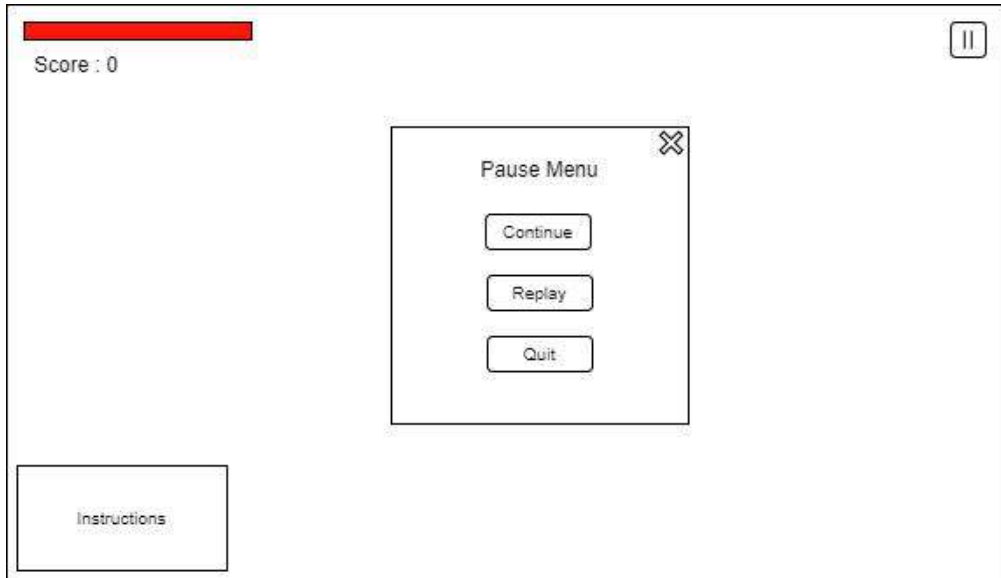
None

PREVIOUS SCENE:

None

Table 3.3 Continued

FRAME 05 (Pause Menu)	
PROJECT TITLE: Found You	SCENE: Game Play Scene



DESCRIPTION:

This is the Pause Menu that allow players to pause the game and decide to continue the game, replay or quit the level and back to Main Menu.

GRAPHIC:

None

BUTTON:

Close Button - close the Pause Menu

Continue - close the Pause Menu

Replay - Restart the level

Quit - save and quit the level, back to Main Menu

AUDIO:

Button Clicking Sound Effect
(paper flip sound)

MUSIC:

None

ANIMATION:

None

PREVIOUS SCENE:

None

3.7 Data Design

3.7.1 ERD Diagram

Figure 3.10 shows the ERD Diagram of the Project. There are two entities which are Player and System. The primary keys of the Player entity are ID and score, while the primary key of the System entity is systemID. The System entity will get the score from players and create the high score. The relationship between the Player and System is one to many, in which one player is able to play the game many times.

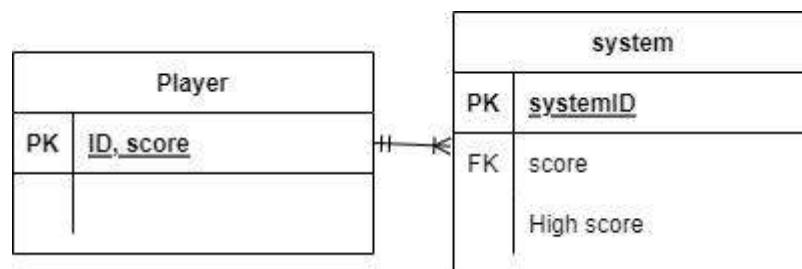


Figure 3.10 ERD Diagram of the Project

3.7.2 Data Dictionary

Table 3.4 Data Dictionary of the Player Entity

Column	Data Type	Description
ID	int	primary key, the unique id of the player
score	int	primary key, the score gain by the player

Table 3.5 Data Dictionary of the System Entity

Column	Data Type	Description
systemID	int	primary key, the unique id of the system
score	int	foreign key, the score that bring from player entity
high score	int	the highest score of the game

3.8 Interface Design

3.8.1 Main Menu

Figure 3.11 shows the interface design of the Main Menu. There are three buttons provided at the centre of the Main Menu, which are New Game, Continue and Quit. These buttons help the player move to another scene, which the New Game is to navigate to the Level 1, Continue is to navigate to previous level and Quit is to exit the application.



Figure 3.11 Interface Design of the Main Menu

3.8.2 Game Interface

Figure 3.12 shows the interface design of the Game Interface. On the top left corner will display the player's health bar and the score. On the top right corner has a pause button, which allows the player to click it to pop up the Pause Menu on the centre of the interface. The instructions of the game will be shown on the bottom left corner of the Game Interface.

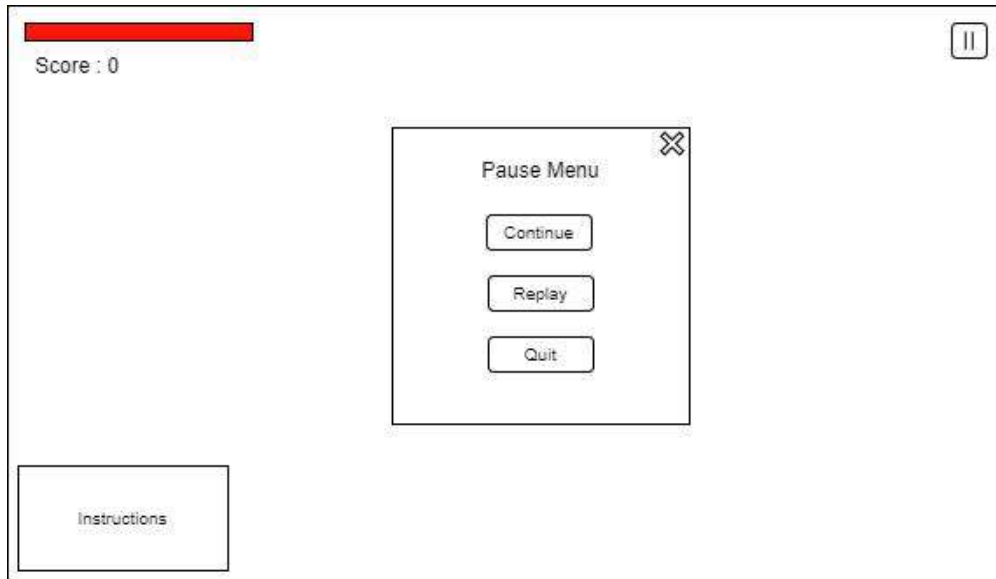


Figure 3.12 Interface Design of the Game Interface

3.8.3 Pause Menu

Figure 3.13 shows the interface design of the Pause Menu. There are four buttons provided which are close, Continue, Replay, and Quit. The close button and Continue are to allow the player to close the Pause Menu and back to the game, Replay is to restart the Level, while Quit allows the user to quit the game and navigate to the Main Menu.

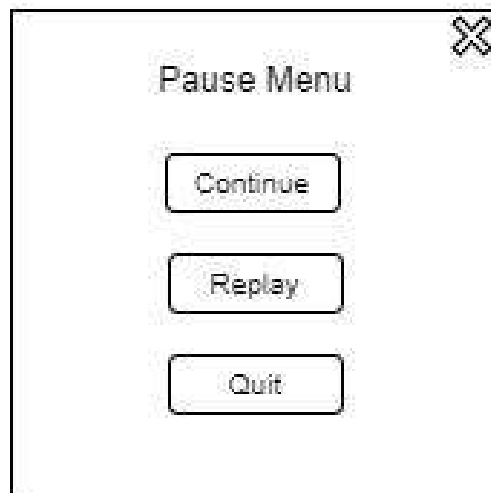


Figure 3.13 Interface Design of the Pause Menu

3.8.4 Lose Panel

Figure 3.14 shows the interface design of the Lose Scene. The scene will be displayed once the player's health bar becomes empty. There are two buttons provided which are Replay for the player to replay the level and Quit for the player to quit the game and navigate to the Main Menu.

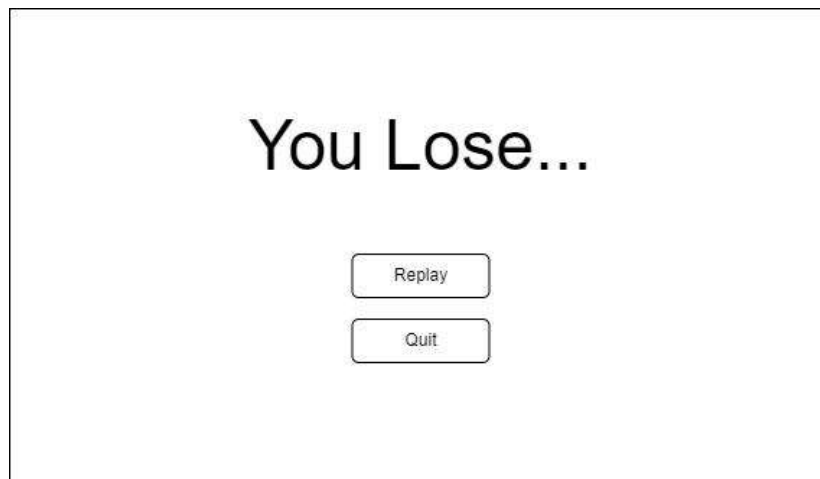


Figure 3.14 Interface Design of the Lose Scene

3.8.5 Win Scene

Figure 3.15 shows the interface design of the Win Scene. Players are able to view their score and the high score in the scene. Besides, the Main Menu button is provided to allow players to navigate to the Main Menu.



Figure 3.15 Interface Design of the Win Scene

3.9 Testing Plan

Table 3.6 is the plan of the UAT for the project that will be released to end users to test the performance of the game. There are 33 activities which are divided into four modules to be tested. The Main Menu, Pause Menu and Lose Scene modules are to test the buttons performance. The Game Level module is to test the player movement, enemy's and NPC's behaviour, checkpoint and pause button. The Win Scene module is to test the display of the total score and high score, and the performance of the Main Menu button. End users only need to play the game and write down the actual result. If the actual result is the same as the expected result, then the status for the activity is passed, else it fails and the developer has to fix the problem.

Table 3.6 UAT (User Acceptance Test) Plan of the Project

No	Module	Activity	Expected Result	Actual Result	Status (Pass / Fail)
1		Display the Main Menu interface	Able to view the Main Menu interface		
2	Main Menu	Click the New Game button	Navigate to the game		
3		Click the Continue button	Navigate to previous stopped level		
4		Click the Quit button	Quit the game application		
5		Press W when on the bottom of a ladder	Player moves up		
6		Press S when on the top of a ladder	Player moves down		
7	Game Level	Press A	Player moves left		
8		Press D	Player moves right		
9		Press E without key	Door cannot open		
10		Press E with key	Door open		

Table 3.6 Continued

No	Module	Activity	Expected Result	Actual Result	Status (Pass / Fail)
11		Press Space when collide with the searchable furnitures	The furniture change the animation to “isOpen”, pop out the collected item, score + 2		
12		Press Space when Collide with grabbable objects	hold the objects		
13		Press Space when Collide with Notes	show notes		
14		Press Space when hold grabbable objects	throw the objects		
15		display the score on the top left of the scene	Able to view the score		
16		display the player’s health bar on the top left of the scene	Able to view the health bar		
17	Game Level	Collide with enemy	player’s health bar - 5		
18		throw object to enemy	enemy play die animation and disappear, score + 5		
19		Health bar become empty	Display the Lose Scene		
20		Get bandage	health bar + 20		
21		Click the Pause button	pop up the Pause Menu		
22		Collide with NPC	NPC display dialogue		
23		Collide checkpoint	Navigate to next level		
24		Collide the last level checkpoint	Display the Win Scene		

Table 3.6 Continued

No	Module	Activity	Expected Result	Actual Result	Status (Pass / Fail)
25		Click the close button	close the Pause Menu		
26	Pause Menu	Click the Continue button	close the Pause Menu		
27		Click the Replay button	Replay the level		
28		Click the Quit button	Navigate to Main Menu		
29	Lose Scene	Click the Replay button	Replay the level		
30		Click the Quit button	Navigate to the Main Menu		
31	Win Scene	Display the total score	Able to view the total score		
32		Display the high score	Able to view the high score		
33		Click the Main Menu button	Navigate to the Main Menu		

3.10 Gantt Chart

Figure 3.16 shows the Gantt chart of the project that refers to the Spiral Model. The green colour is the complete part, purple colour is in progress and the grey colour is to-do part. Based on the figure, the completed parts are Planning, Design, Implement phase for Cycle 1. The in-progress parts is the Play Test phase, while the to-do parts are Evaluate phase for Cycle 1, Planning & design, Implement, Play test and Evaluate phase for Cycle 2 and 3.

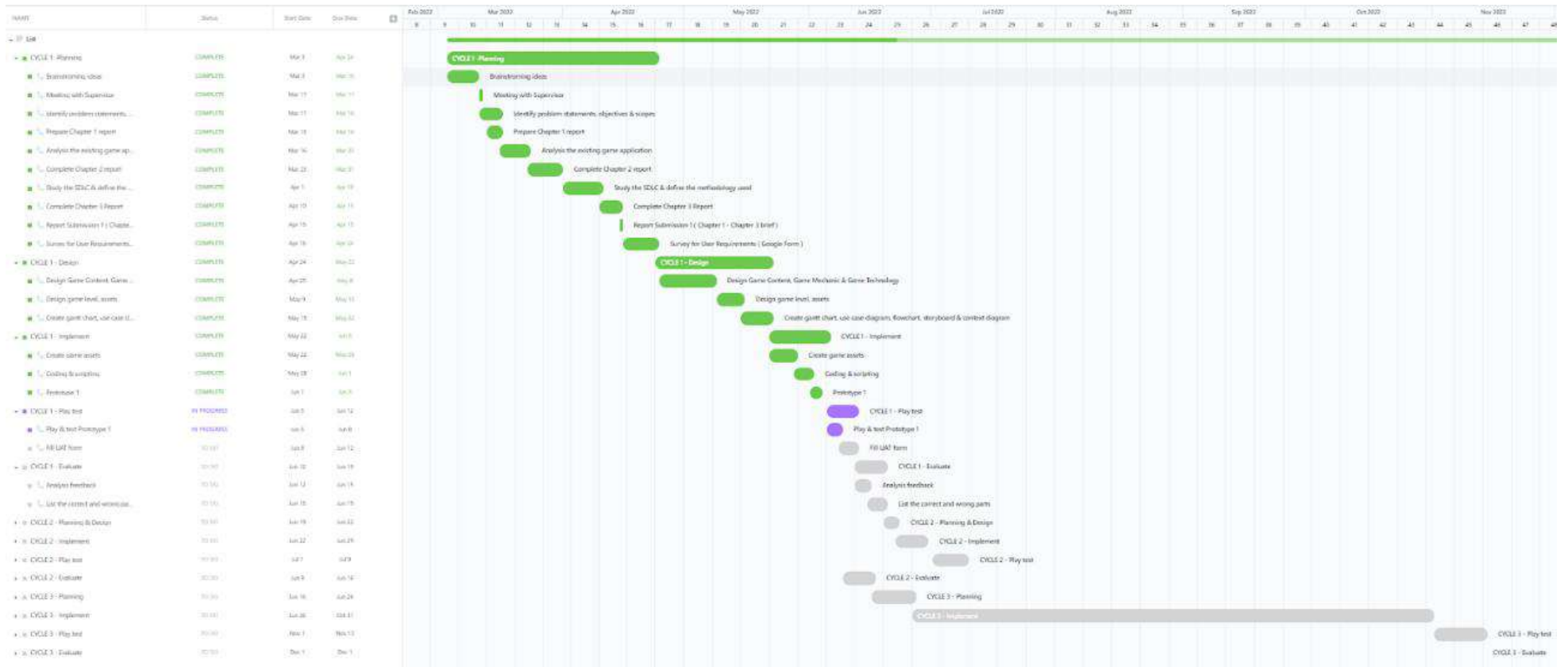


Figure 3.16 Gantt Chart of the Project

3.11 Potential Use of Proposed Solution

Found You is able to improve players' cognitive ability which players must think and choose carefully when encountering kidnappers to prevent the character from being kidnapped in the game. Thus, players are able to learn what they must do when encountering a kidnapper.

Next, Found You improves players' problem-solving skills such as analysis and communication. In the game, players learn to analyse the message that they get from the NPCs in the game to predict the story. Players will learn the importance of communication.

Besides, Found You has a high tendency to help players increase their awareness of kidnapping rather than reading articles, plus there are not many existing games about kidnapping. Therefore, Found You 2D game has the marketing potential to attract public attention.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter is about how the Found You 2D game was implemented by using Unity game engine, Microsoft Visual Studio 2017 and Adobe Photoshop. All the game assets are created using Adobe Photoshop, while all the codes are written using Microsoft Visual Studio.

4.2 Implementation Phase

The details of work using Unity, Visual Studio and Adobe Photoshop will be explained in this Implementation Phase. The game assets are created by Adobe Photoshop and imported to Unity. After that, write the codes using Microsoft Visual Studio to create the C# scripts for development purposes.

4.2.1 Game Assets (Adobe Photoshop)

All the game assets are drawn by using Adobe Photoshop, where the game assets are game objects, characters, environments, UI assets and also animations. Adobe Photoshop allows users to draw their own assets easily where users are able to use different brushes, coloring, editing and applying effects.

Figure 4.1 shows the interface of Adobe Photoshop. On the left of the interface is a toolbar to allow users draw, erase, crop etc. Middle of the interface is the place for users to create their assets. On the right of the interface, there are navigators, color, brushes and layers.

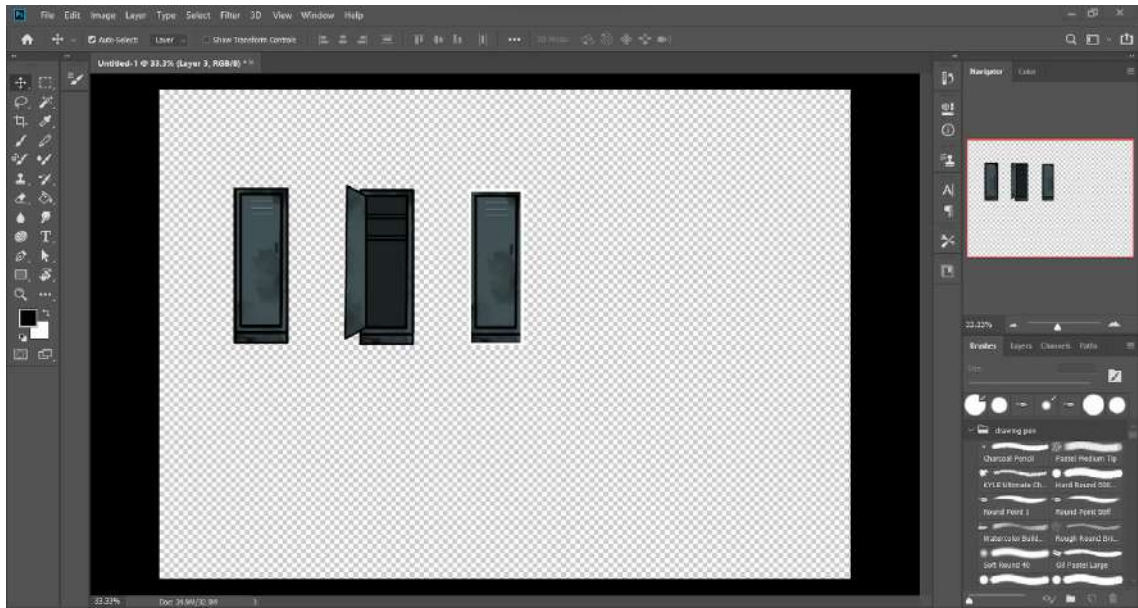


Figure 4.1 Interface of the Adobe Photoshop

Figure 4.2 shows the drawing of the game asset's outline. The outline is using a brush with black color to draw. The type of the brush is Hard Round with a size of 1px.

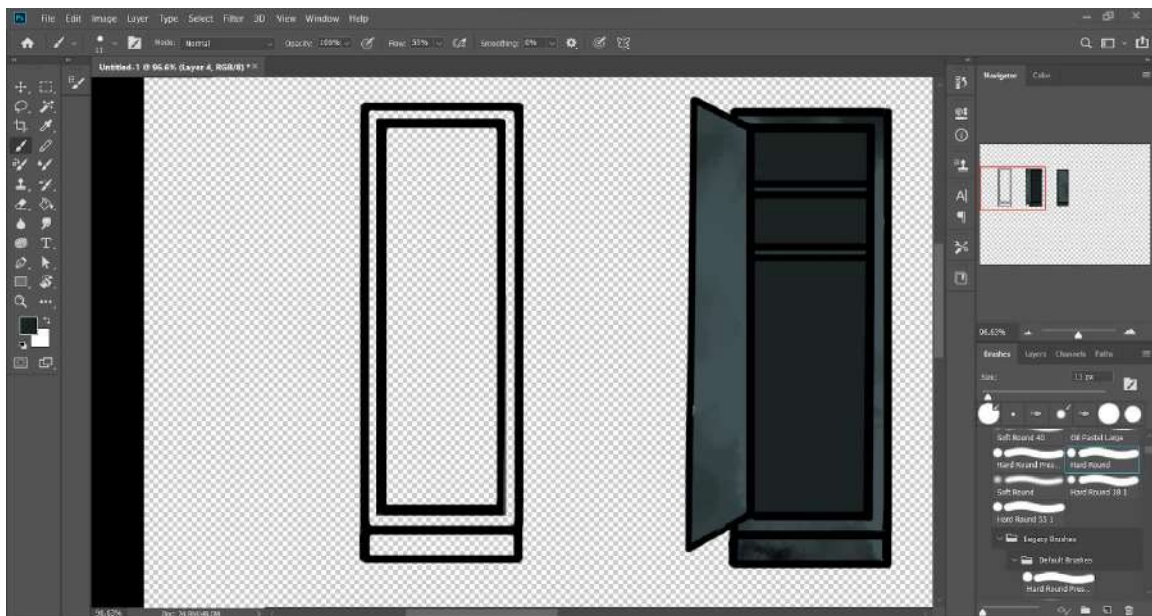


Figure 4.2 Draw outline of game asset

Figure 4.3 shows the color of the game asset. The coloring of the game asset is very easy, where I select a Hard Round Pressure Opacity, choose a color and color it. Then the details are drawn using a brush and the method is the same as coloring.

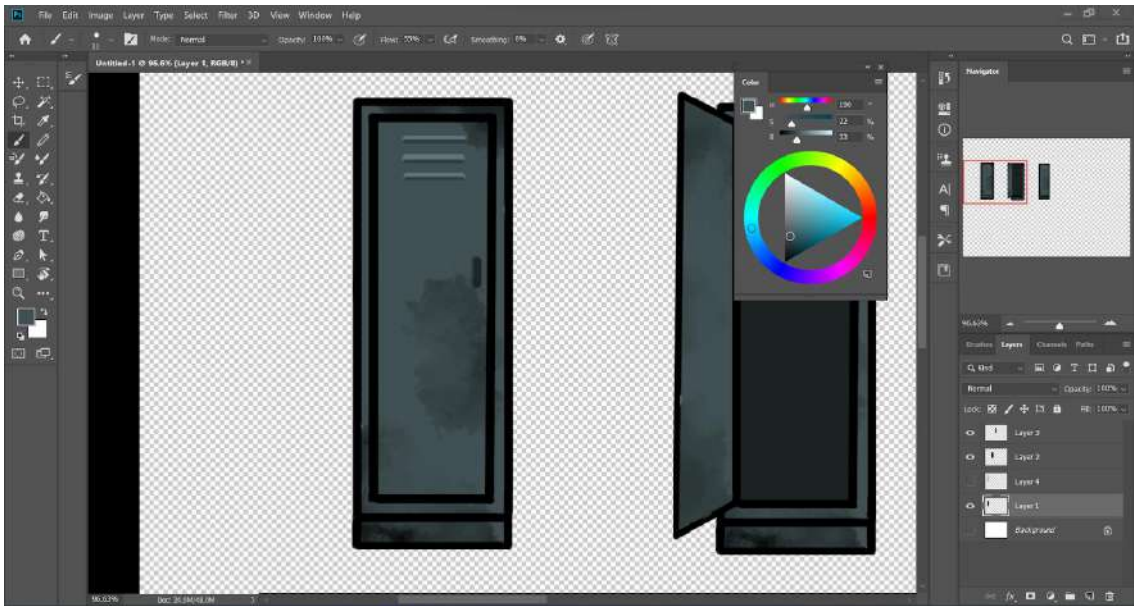


Figure 4.3 Color the game asset

Figure 4.4 shows the step to export the game asset in PNG. After completing creating the game asset, export the asset as PNG and the asset will store on Desktop. The asset that exports is used to import into Unity for the game development.

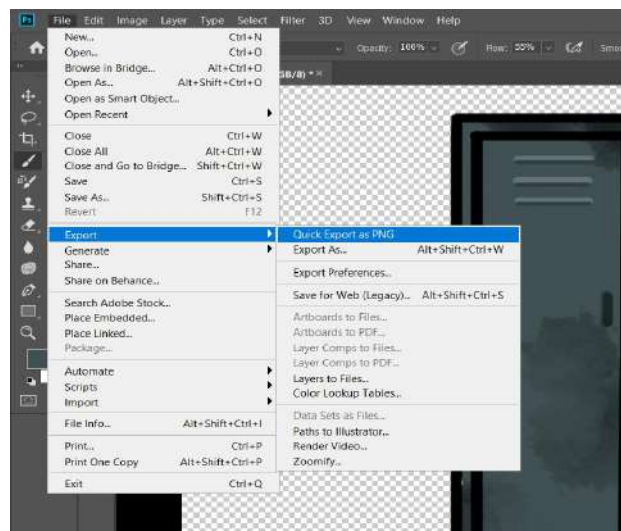


Figure 4.4 Export game asset as PNG

Figure 4.5 shows the created game assets which are drawn by using Adobe Photoshop. These game assets are imported into Unity for game development purposes.



Figure 4.5 Created game assets

Figure 4.6 shows the creation of characters, enemies and animation. The animation of the Main Character (Anne) Enemies (Wolf and Rat) is drawn frame by frame, then will be used in Unity Animation.

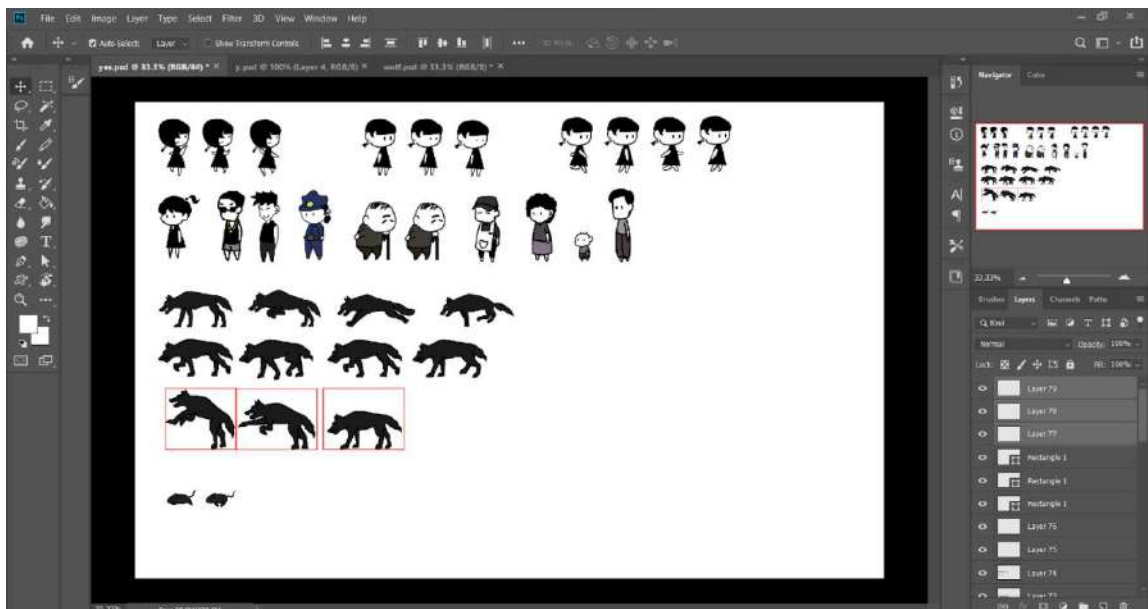


Figure 4.6 Create Characters, Enemies and Animation

4.2.2 Main Menu

Figure 4.7 shows the Main Menu of the game application. The title of the game is shown on the left top of the screen which is “FOUND YOU”. There are 3 buttons provided which are New Game, Continue, Highscore and Quit.

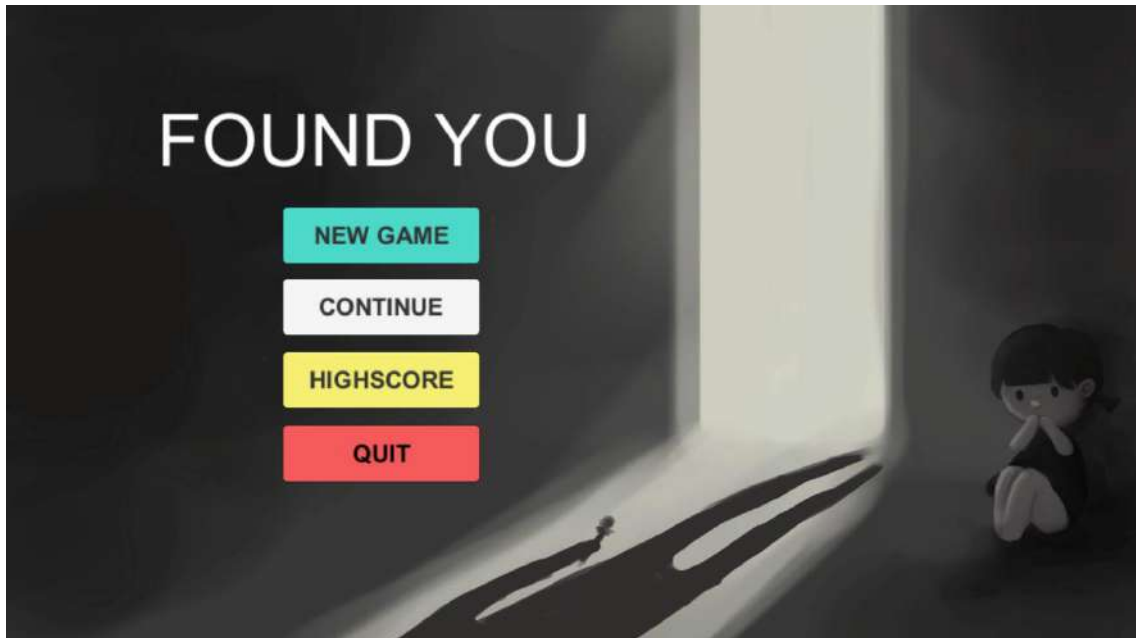


Figure 4.7 Main Menu Interface

Figure 4.8 shows the Inspector of the buttons in the Main Menu. The buttons are used on click () which with the ChangeScene_highscore script to allow the button to navigate to the correct scene after the players click the button.

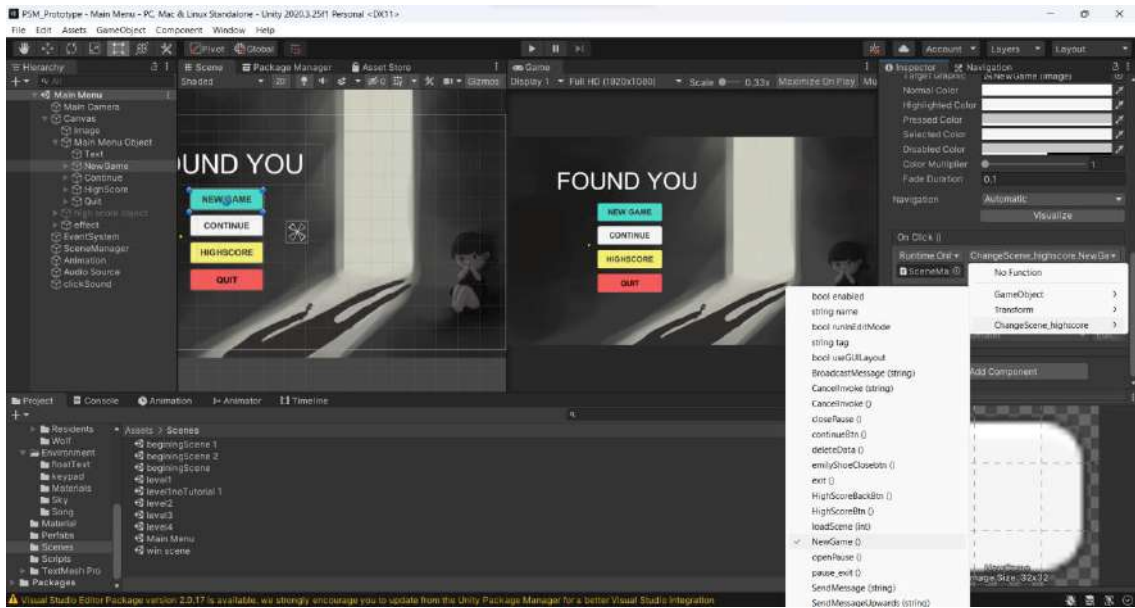


Figure 4.8 Inspector of the Main Menu Buttons

Figure 4.9 shows a part of `ChangeScene_highscore` script that used in Main Menu. When players click the NEW GAME button on the Main Menu, the script will run the `public void NewGame()` and `bool isNewGame` is equal to true, then in `private void Update()`, if `bool isNewGame` is equal to true then load the Game Beginning Instruction scene and delete the saved `PlayerPrefs` for the score and player's health.

When players click the CONTINUE button on the Main Menu, the script will run the `public void continueBtn()`, where if the saved scene is not equal to zero, then it will load the scene index that saved in `PlayerPrefs.GetInt("SavedScene")`.

When players click the HIGHSCORE button on the Main Menu, the script will run the `public void HighScoreBtn()`, which it will hide the Main Menu panel and show the Highscore panel in a same scene.

When players click the QUIT button, the script will run the `public void exit()`, where it will quit the game application.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  using UnityEngine.UI;
6
7  public class ChangeScene_highscore : MonoBehaviour
8  {
9      public GameObject panel;
10     public GameObject emptyShoePanel;
11     public GameObject emptyShoe;
12     public GameObject emptyShoeInstruction;
13     public GameObject TimelineGameBegin;
14     public GameObject lovePanel;
15
16     public GameObject mainMenuUI;
17     public GameObject highscoreUI;
18
19     public Text score;
20     public Text newhighscore;
21     public Text highscore;
22
23     private int currentSceneIndex;
24     private int sceneToContinue;
25
26     public AudioSource clickSound;
27
28     public bool open = false;
29     public bool isNewGame = false;
30
31     public float timer;
32
33     private void Start()
34     {
35         score.text = "SCORE : " + PlayerPrefs.GetInt("currentscore");
36         newhighscore.text = "NEW HIGH SCORE : " + PlayerPrefs.GetInt("newhighscore");
37         highscore.text = "HIGH SCORE : " + PlayerPrefs.GetInt("allhighscore");
38     }
39
40     private void Update()
41     {
42         if (Input.GetMouseButton(0))
43         {
44             clickSound.Play();
45
46             if (isNewGame)
47             {
48                 timer -= Time.deltaTime;
49                 if (timer <= 0)
50
51                     {
52                         timer = Time.deltaTime;
53                     }
54                 SceneManager.LoadScene(7);
55                 open = false;
56                 PlayerPrefs.DeleteKey("savedscore");
57                 PlayerPrefs.DeleteKey("savedscore");
58                 PlayerPrefs.DeleteKey("savedscore");
59                 PlayerPrefs.DeleteKey("savedscore");
60
61                 PlayerPrefs.DeleteKey("healthlv1");
62                 PlayerPrefs.DeleteKey("healthlv2");
63                 PlayerPrefs.DeleteKey("healthlv3");
64                 PlayerPrefs.DeleteKey("healthlv4");
65                 isNewGame = false;
66             }
67         }
68         if (open)
69         {
70             timer.timescale = 0;
71         }
72         else
73         {
74             timer.timescale = 1;
75         }
76     }
77
78     public void NewGame()
79     {
80         clickSound.Play();
81         isNewGame = true;
82     }
83
84     public void loadScene(int index)
85     {
86         clickSound.Play();
87         SceneManager.LoadScene(index);
88         open = false;
89     }
90
91     public void exit()
92     {
93         clickSound.Play();
94         Application.Quit();
95         Debug.Log("Game is exiting");
96     }
97
98 }

```

Figure 4.9 ChangeScene_highscore Script in Main Menu

4.2.3 Pause Menu

Figure 4.10 shows the Pause Menu of the game application. There are 3 buttons provided which are New Game, Continue and Quit.

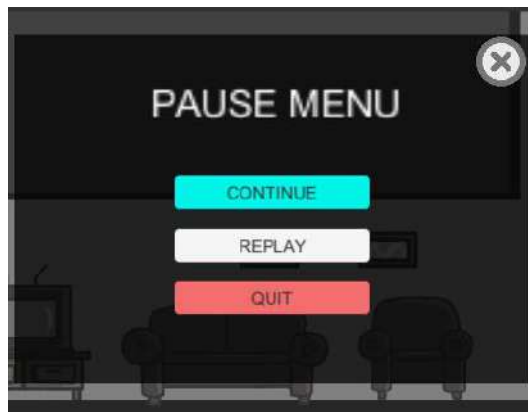


Figure 4.10 Pause Menu Interface

Figure 4.11 shows the Inspector of the buttons in the Pause Menu. The buttons are used on click () which with the ChangeScene script to allow the button to navigate to the correct scene after the players click the button.

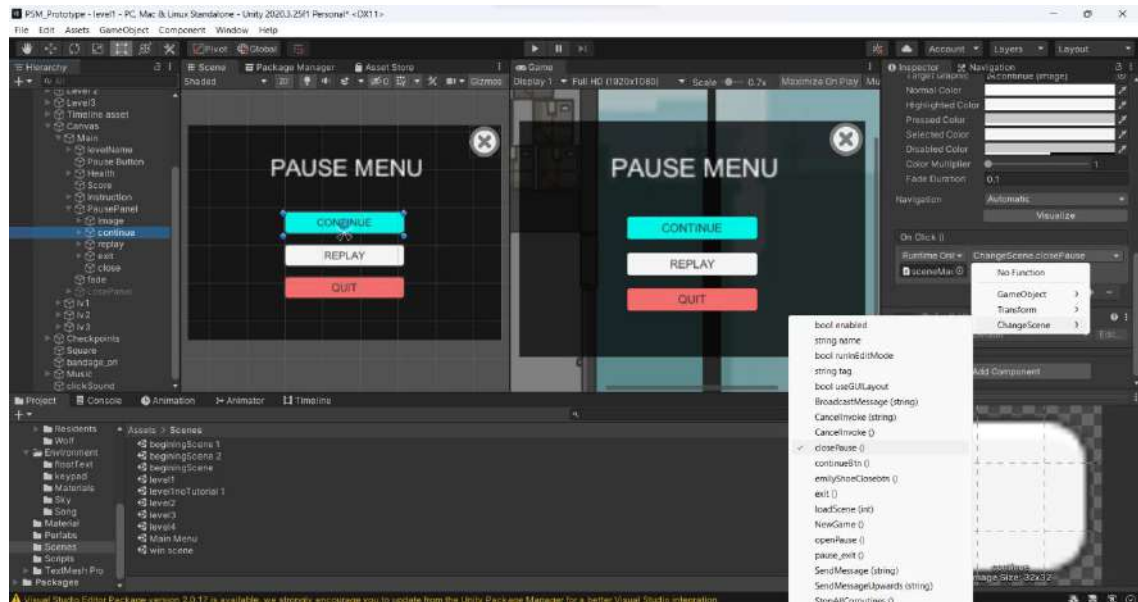


Figure 4.11 Inspector of the Pause Menu Buttons

Figure 4.12 shows the a part of ChangeScene script that will be used in Pause Menu. When players click the Pause icon button on the top right of the screen, the script will rin the *public void openPause()* which it will show the Pause Menu panel. When players click the CONTINUE button or the Close icon button on the Pause Menu, the script will run the *public void closePause()* which it will hide the Pause Menu panel.

When players click the REPLAY button, the script will run the *public void loadScene(int)* which it will reload the Level scene. When players click the QUIT button on the Pause Menu, the script will run the *public void pause_exit()* which the scene index will be saved in *PlayerPrefs.SetInt("SavedScene")*, after that it will load to Main Menu.

```
ChangeScene.cs
Assembly-CSharp
ChangeScene
panel

61
62
63 public void exit()
64 {
65     Application.Quit();
66     Debug.Log("Game is exiting");
67 }
68
69 public void openPause()
70 {
71     clickSound.Play();
72     panel.SetActive(true);
73     open = true;
74 }
75
76 public void closePause()
77 {
78     clickSound.Play();
79
80     panel.SetActive(false);
81     open = false;
82 }
83
84 public void pause_exit()
85 {
86     clickSound.Play();
87
88     currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
89     PlayerPrefs.SetInt("SavedScene", currentSceneIndex);
90     SceneManager.LoadScene(0);
91     open = false;
92 }
93
```

Figure 4.12 ChangeScene Script in Pause Menu

4.2.4 Lose Panel

Figure 4.13 shows the Lose Panel interface of the game application. The Lose Panel will be shown when players' health bar is empty or players select the wrong answer. There are 2 buttons provided in the Lose Panel which are Replay and Quit.

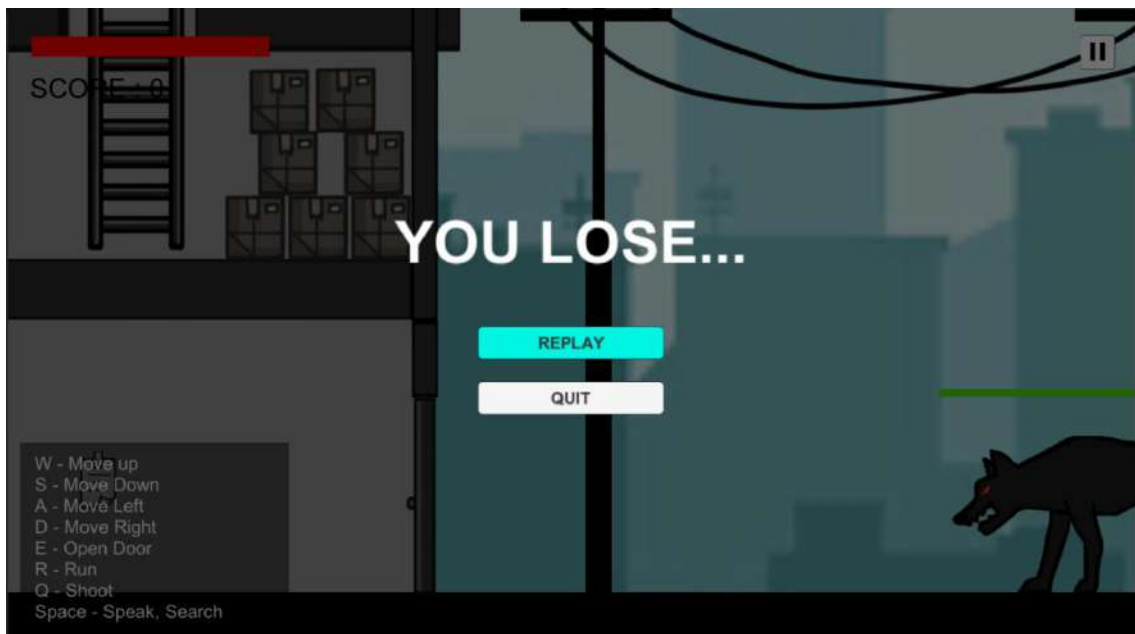


Figure 4.13 Lose Panel Interface

Figure 4.14 shows the Inspector of the buttons in the Lose Panel. The buttons are used on click () which with the ChangeScene script to allow the button to navigate to the correct scene after the players click the button.

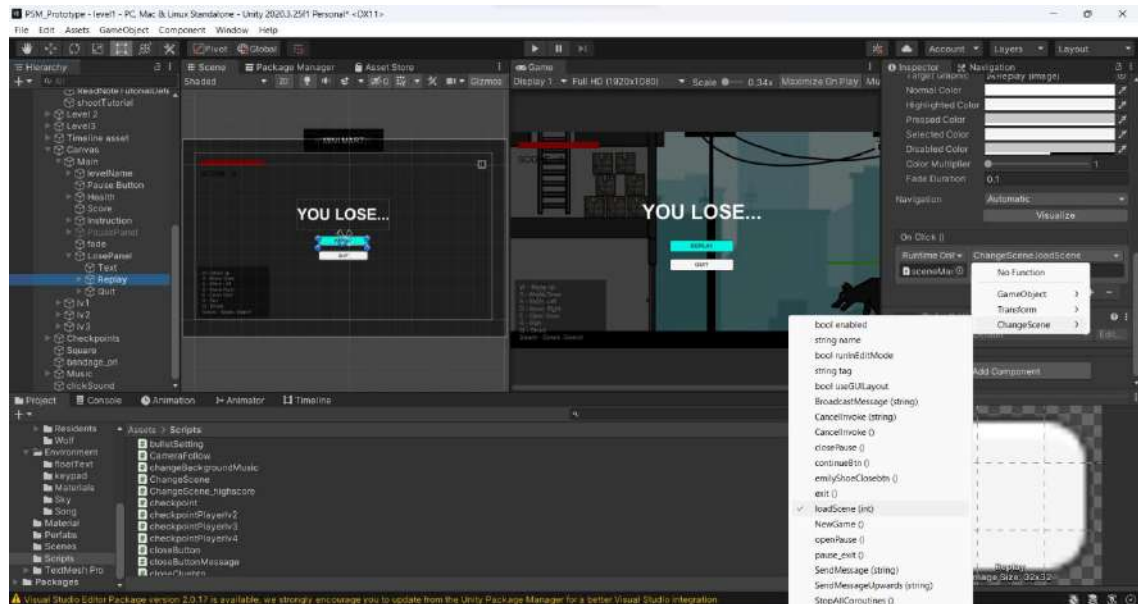


Figure 4.14 Inspector of the Lose Panel Buttons

Figure 4.15 shows the a part of ChangeScene script that will be used in Lose Panel which is similar with Pause Menu.

When players click the REPLAY button on the Lose Panel, the script will run the *public void loadScene(int)* which it will reload the Level scene.

When players click the QUIT button on the Lose Panel, the script will run the *public void pause_exit()* which the scene index will be saved in *PlayerPrefs.SetInt("SavedScene")*, after that it will load to Main Menu.

```
ChangeScene.cs
Assembly-CSharp
ChangeScene
panel

61 }
62
63 public void exit()
64 {
65     Application.Quit();
66     Debug.Log("Game is exiting");
67 }
68
69 public void openPause()
70 {
71     clickSound.Play();
72     panel.SetActive(true);
73     open = true;
74 }
75
76 public void closePause()
77 {
78     clickSound.Play();
79     panel.SetActive(false);
80     open = false;
81 }
82
83
84 public void pause_exit()
85 {
86     clickSound.Play();
87
88     currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
89     PlayerPrefs.SetInt("SavedScene", currentSceneIndex);
90     SceneManager.LoadScene(0);
91     open = false;
92 }
93 }
```

Figure 4.15 ChangeScene Script in Lose Panel

4.2.5 Highscore Panel

Figure 4.16 shows the Highscore Panel of the game application. Highscore Panel will be shown when players click the HIGHSCORE button on the Main Menu. There are 2 buttons provided which are Back and Clear. Players are able to view the current score they gained, new high score and the high score which is the previous high score.

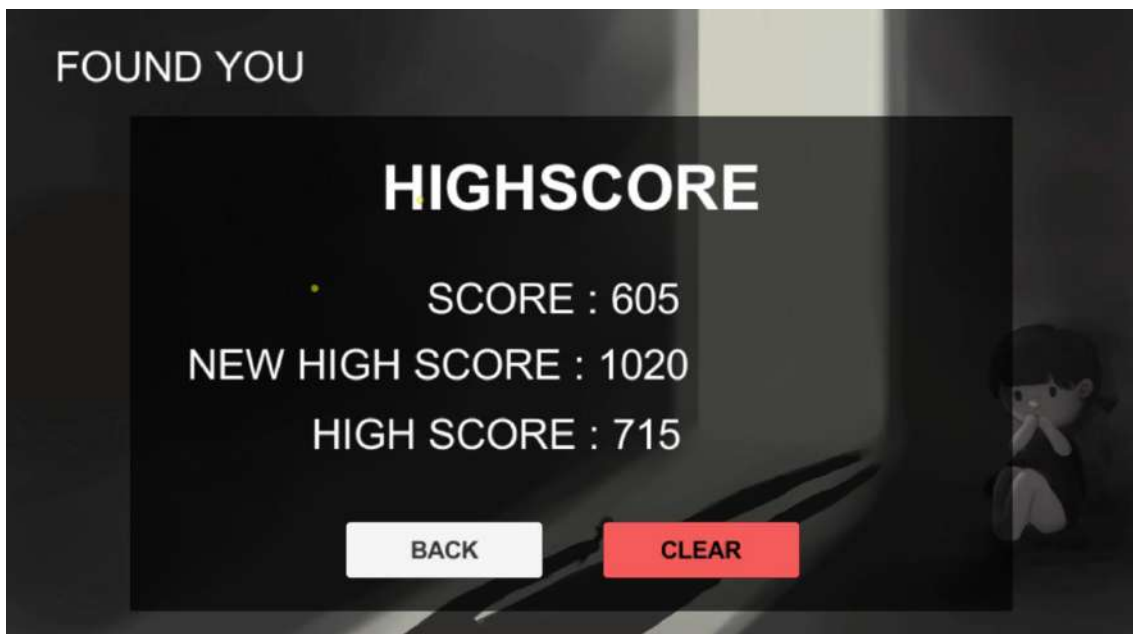


Figure 4.16 Highscore Panel Interface

Figure 4.17 shows the Inspector of the buttons in the Highscore Panel. The buttons are used on click () which with the ChangeScene_highscore script to allow the button to navigate to the correct scene after the players click the button.

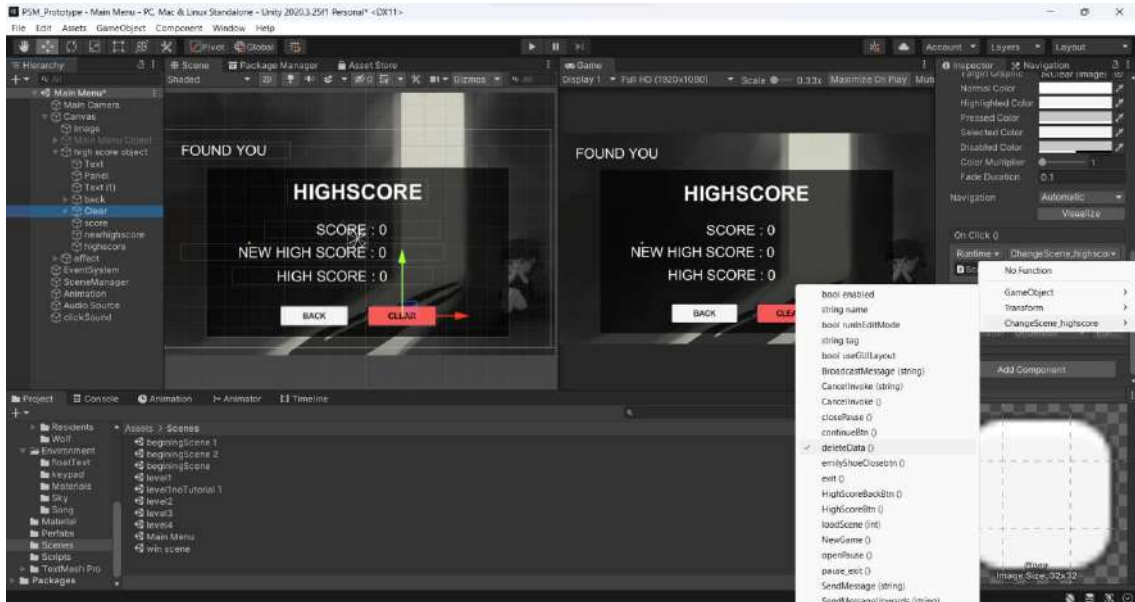


Figure 4.17 Inspector of the Highscore Panel Buttons

Figure 4.18 shows the a part of ChangeScene_highscore script that will be used in Highscore Panel. When players click the BACK button on the Highscore Panel, the script will run the *public void loadScene(int)* which it will reload to the Main Menu.

When players click the CLEAR button on the Highscore Panel, the script will run the *public void deleteData()* which the script will delete the key of the saved current score, old high score, new high score and high score. Then the text of the Score, New High Score and High Score will be zero.


```
ChangeScene_highscore.cs # X
Assembly-CSharp ChangeScene_highscore panel
154 Destroy(emilyShoeInstruction, 3f);
155
156 }
157
158 public void deleteData()
159 {
160     clickSound.Play();
161
162     PlayerPrefs.DeleteKey("oldhighscore");
163     PlayerPrefs.DeleteKey("newhighscore");
164     PlayerPrefs.DeleteKey("currentscore");
165     PlayerPrefs.DeleteKey("highscore");
166
167     score.text = "SCORE : " + PlayerPrefs.GetInt("currentscore");
168     newhighscore.text = "NEW HIGH SCORE : " + PlayerPrefs.GetInt("newhighscore");
169     highscore.text = "HIGH SCORE : " + PlayerPrefs.GetInt("oldhighscore");
170 }
171
172
173 public void HighScoreBtn()
174 {
175     clickSound.Play();
176
177     mainMenuUI.SetActive(false);
178     highScoreUI.SetActive(true);
179 }
180
181 public void HighScoreBackBtn()
182 {
183     clickSound.Play();
184
185     mainMenuUI.SetActive(true);
186     highScoreUI.SetActive(false);
187 }
188
189 }
```

Figure 4.18 ChangeScene_highscore Script in Highscore Panel

4.2.6 Game Beginning Instruction

Figure 4.19, Figure 4.20 and Figure 4.21 show the Game Beginning Instructions of the game application. These instructions will be shown after players click the NEW GAME button. First will shows the Game Beginning Instruction 1, then follow by Game Beginning Instruction 2 and 3.

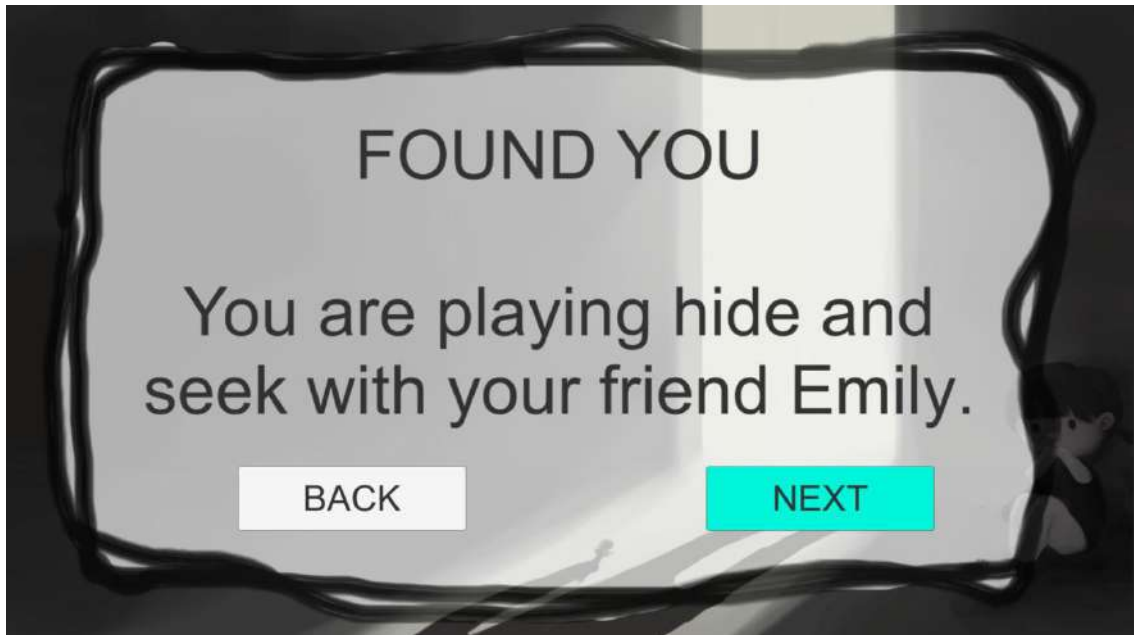


Figure 4.19 Game Beginning Instruction 1 Interface



Figure 4.20 Game Beginning Instruction 2 Interface

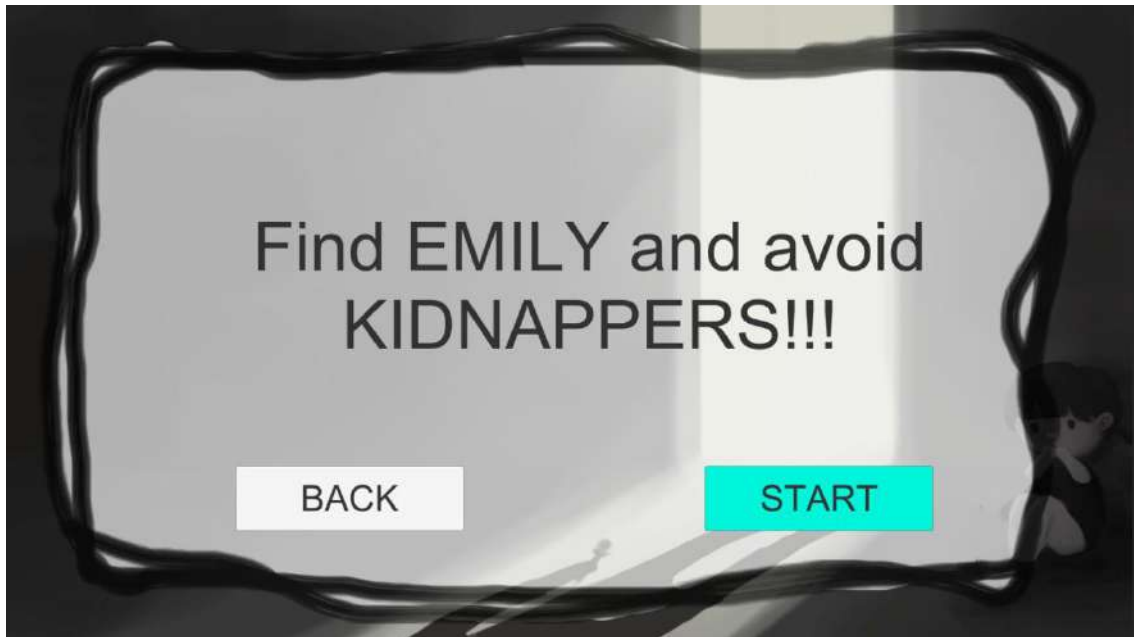


Figure 4.21 Game Beginning Instruction 3 Interface

Figure 4.22, Figure 4.23 and Figure 4.24 show the Inspector of the buttons in the Highscore Panel. The buttons are used on click () which with the ChangeScene_highscore script to allow the button to navigate to the correct scene after the players click the button.

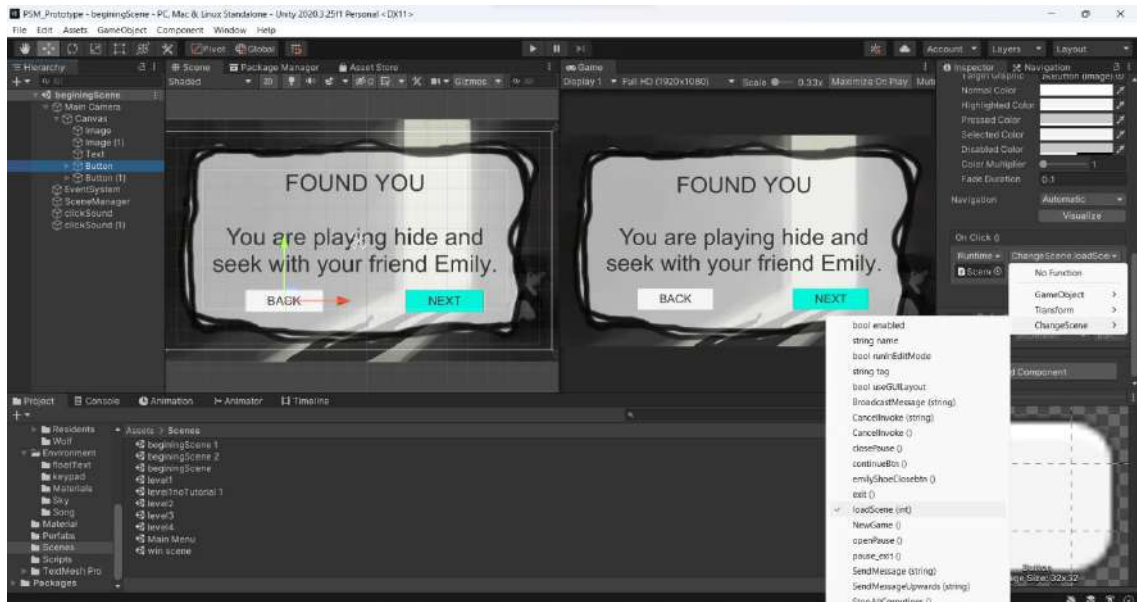


Figure 4.22 Inspector of the Game Beginning Instruction 1 Buttons

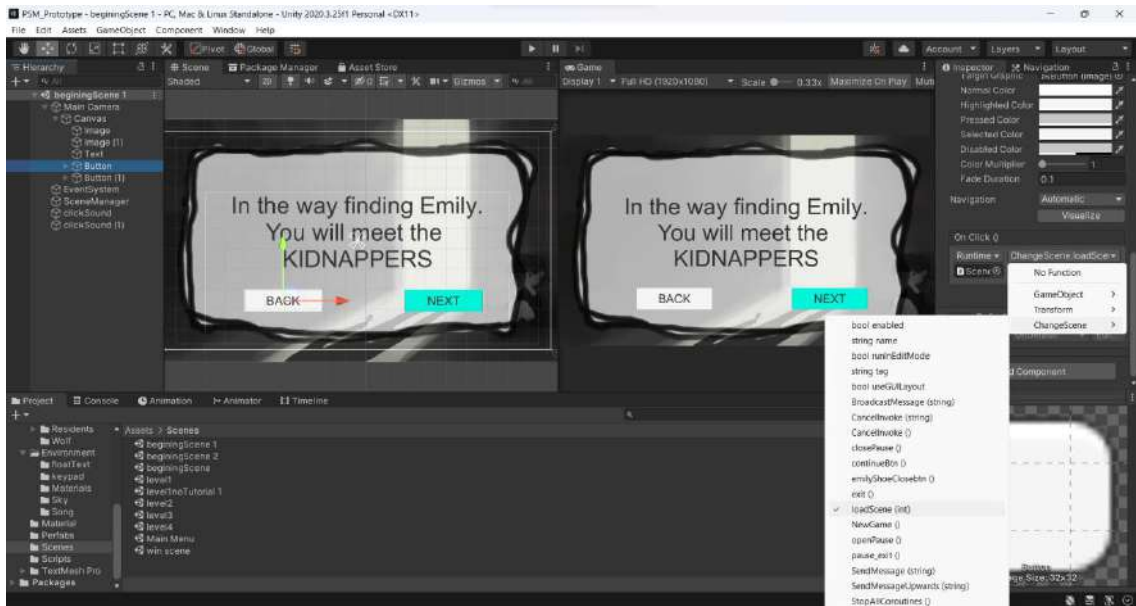


Figure 4.23 Inspector of the Game Beginning Instruction 2 Buttons

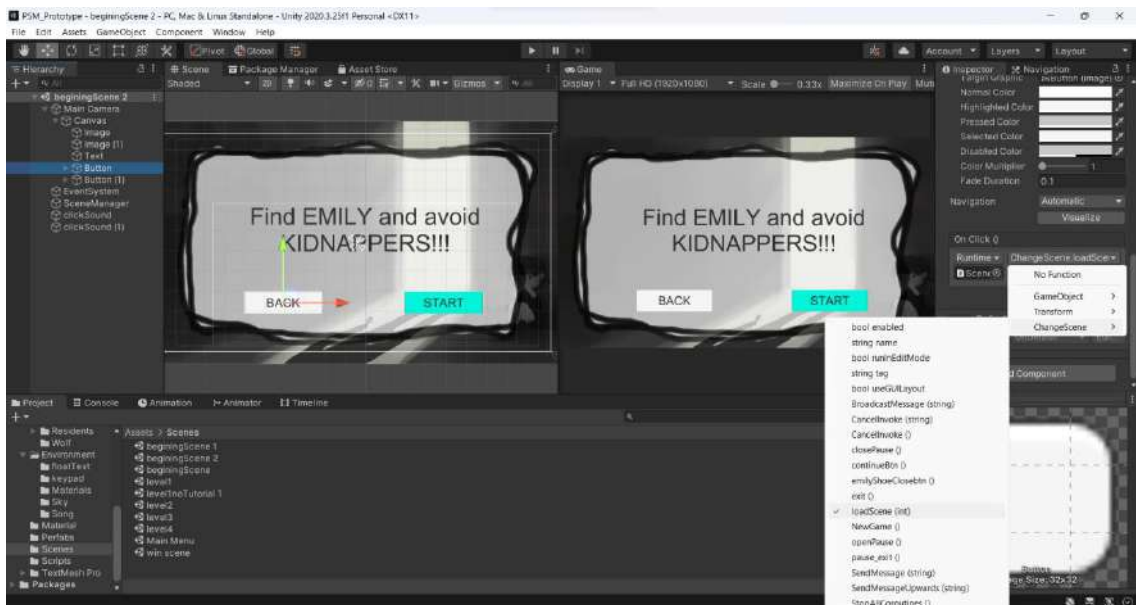
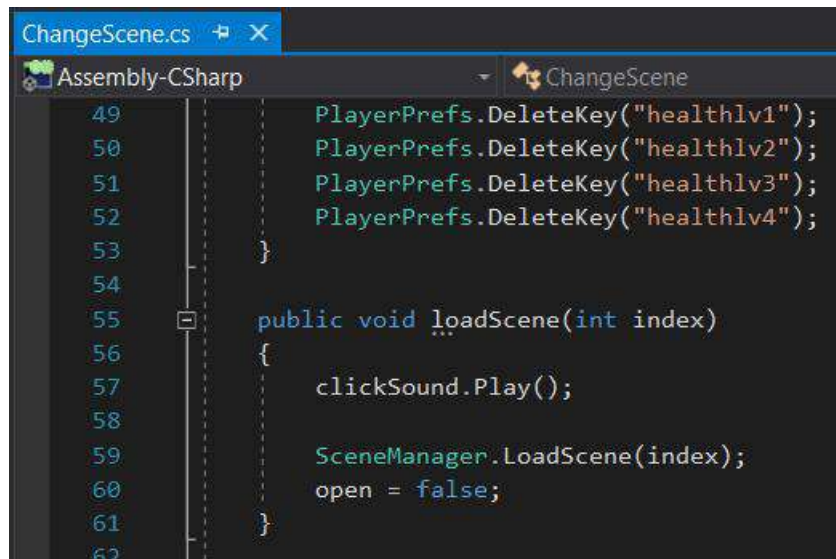


Figure 4.24 Inspector of the Game Beginning Instruction 3 Buttons

Figure 4.25 shows a part of the `ChangeScene_highscore` script that will be used in Game Beginning Instruction. When players click the BACK button on the Game Beginning Instruction 1, the script will run the `public void loadScene(int)` which will reload to the Main Menu. While the BACK button in Game Beginning Instruction 2 will go back to Game Beginning Instruction 1.

When players click the NEXT button on the Game Beginning Instruction 1 , the script will run the *public void loadScene(int)* which it will reload to the Game Beginning Instruction 2. While the NEXT button in Game Beginning Instruction 2 will go to Game Beginning Instruction 3.

When players click the START button on the Game Beginning Instruction 3, the script will run the *public void loadScene(int)* it will load to the Level 1.



```
ChangeScene.cs + X
Assembly-CSharp ChangeScene
49 PlayerPrefs.DeleteKey("healthlv1");
50 PlayerPrefs.DeleteKey("healthlv2");
51 PlayerPrefs.DeleteKey("healthlv3");
52 PlayerPrefs.DeleteKey("healthlv4");
53 }
54
55 public void loadScene(int index)
56 {
57     clickSound.Play();
58
59     SceneManager.LoadScene(index);
60     open = false;
61 }
62
```

Figure 4.25 ChangeScene Script in Game Beginning Instruction

4.2.7 Game User Interface

Figure 4.26 shows the game user interface and Figure 4.27 shows the game user interface development in Unity. On the top left of the interface is the Score that shows the score obtained by players along the game, bottom left is a player movement instruction, and top right is a Pause button which is showing the Pause Menu after the player clicks the button.

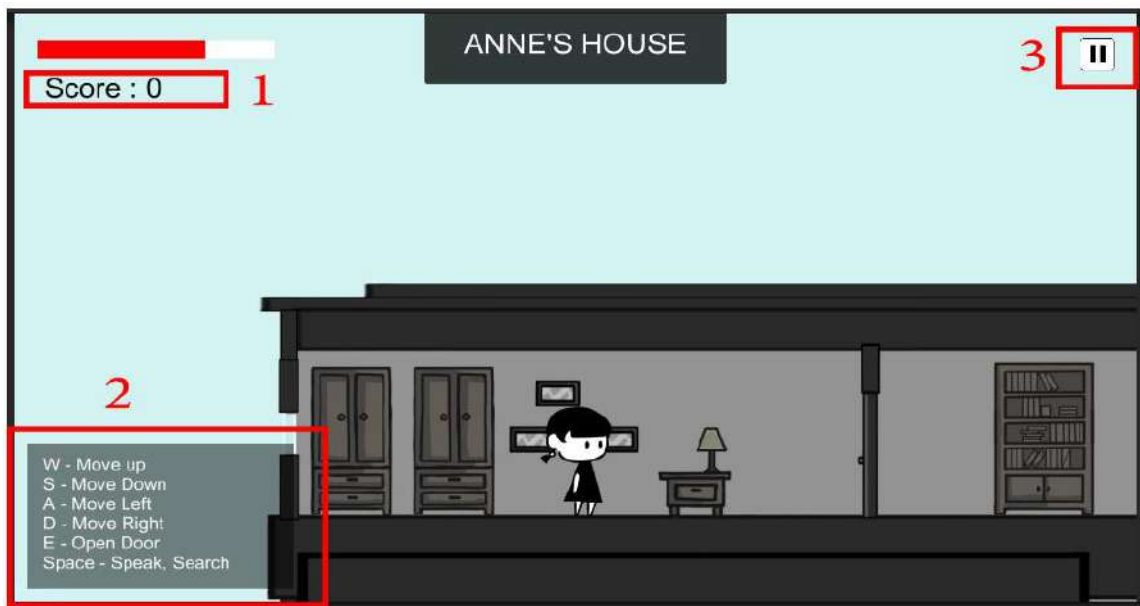


Figure 4.26 Game User Interface

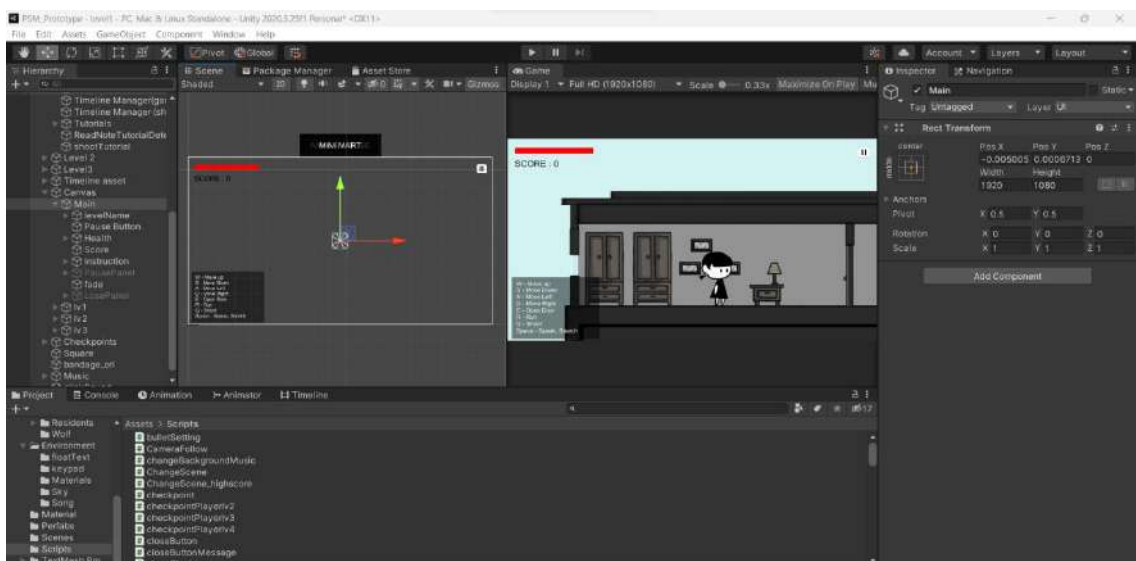


Figure 4.27 Game User Interface Development in Unity

4.2.8 Game Level Interface

Level 1 - Anne's House



Figure 4.28 Game Environment Design of Level 1

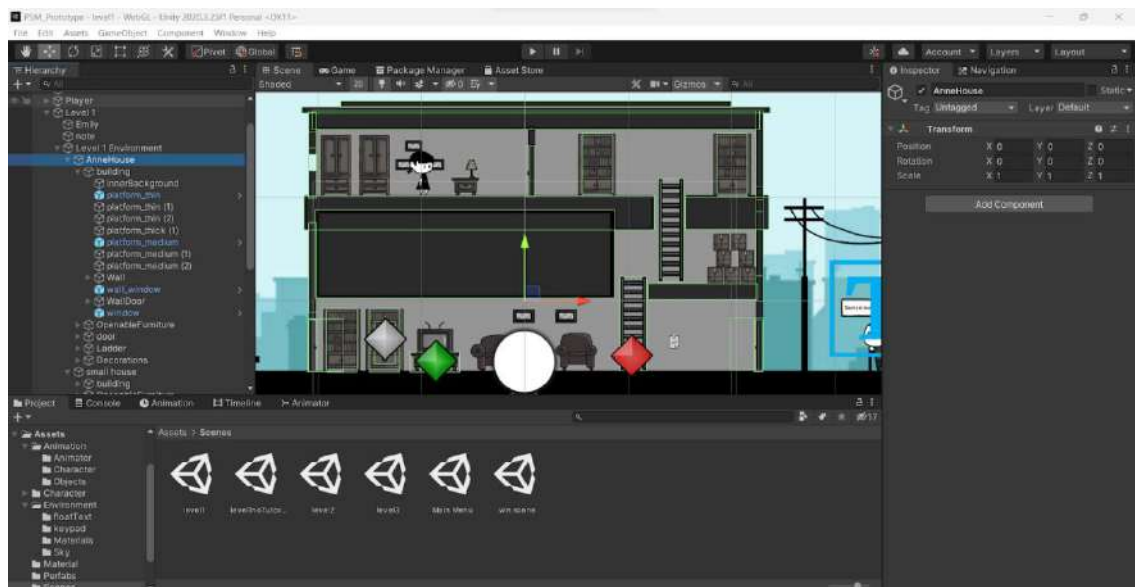


Figure 4.29 Level 1 Anne's House Development

Figure 4.30 shows the timeline for Game Begin development in Level 1. This timeline will play on awake once players load the level 1 scene. This timeline is about Anne's friend, Emily, who comes to her room and asks to play hide and seek with her. The `lvlgamebegin` is to start the tutorial 1 which teach players to move.

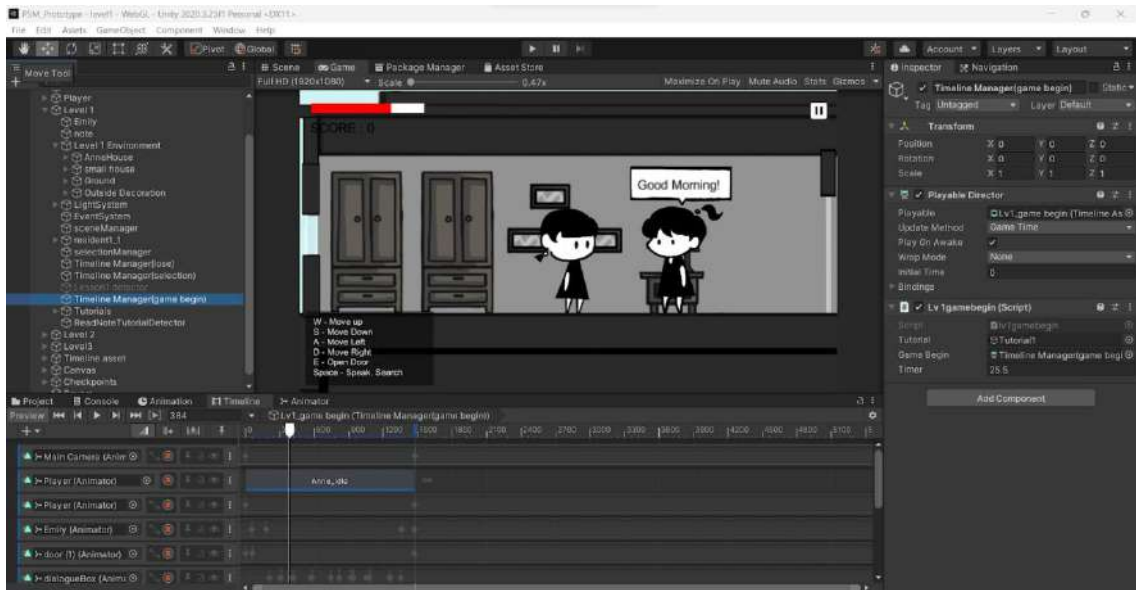


Figure 4.30 Timeline of the Game Begin in Level 1

Figure 4.32 shows the `lv1gamebegin` Script. The `GameObject` tutorial is the Tutorial 1 (Figure 4.31), the `PlayableDirector` is the Timeline Manager(Game Begin) that obtained the timeline game begin, and the timer is to countdown the time of the timeline finished with initial 25.5 seconds. When the game is started, the timer will start a countdown of 25.5 seconds. Once the timer is less or equal to zero, the script will stop the timeline and show the Tutorial 1.

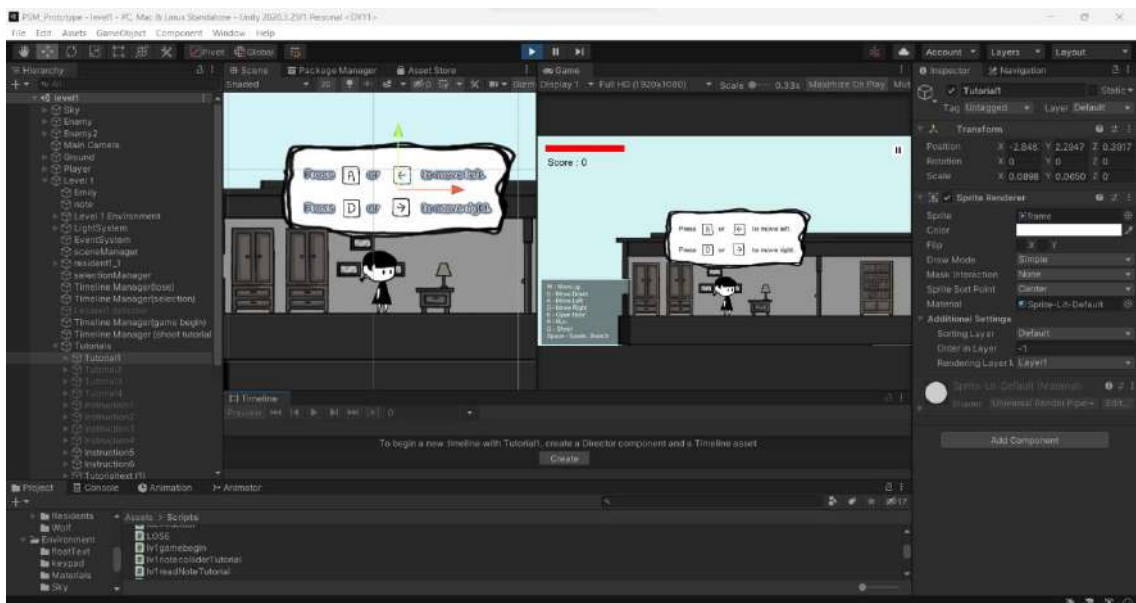


Figure 4.31 Tutorial 1 Player Movement Development


```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.Playables;
6
7  public class lv1gamebegin : MonoBehaviour
8  {
9      public GameObject tutorial;
10     public PlayableDirector gameBegin;
11     public float timer;
12     private bool startcount = true;
13
14     void Start()
15     {
16
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22         if(startcount)
23         {
24             timer -= Time.deltaTime;
25             if (timer <= 0)
26             {
27                 tutorial.SetActive(true);
28                 gameBegin.Stop();
29                 startcount = false;
30             }
31         }
32     }
33 }
34

```

Figure 4.32 lv1gamebegin Script in Tutorial 1

Figure 4.33 shows the Tutorial 1 in the Level 1 of the game application. Tutorial 1 is to teach players how to move the Anne by pressing AD or left right arrow buttons.

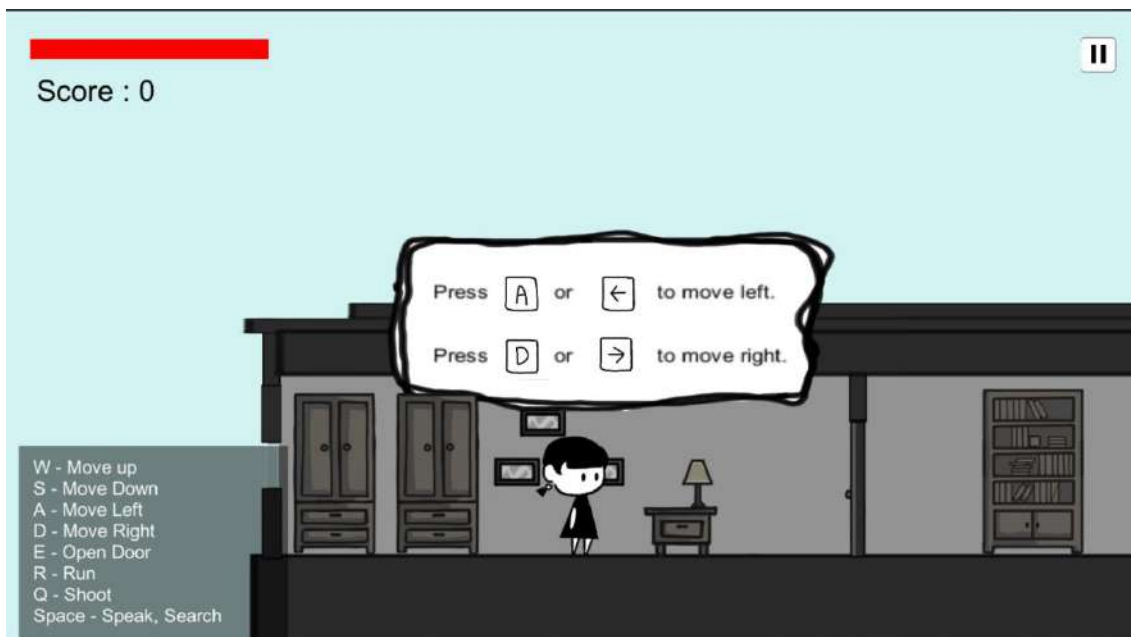


Figure 4.33 Interface of the Tutorial 1 In Level 1

Figure 4.34 shows the lv1tutorial1To2 Script in Tutorial 1. When the Tutorial 1 panel shown and players press any button from AD or left right arrow buttons, the float timer will start count down for 5 seconds. After 5 seconds, Tutorial 1 panel will be hidden and show the Tutorial 2 panel.

```

lv1tutorial1To2.cs
Assembly-CSharp
lv1tutorial1To2
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class lv1tutorial1To2 : MonoBehaviour
6 {
7     public float walkspeed;
8     public float climbspd;
9     private Animator anim;
10    public Rigidbody2D rb;
11    public GameObject walktutorial;
12    public GameObject tutorial2;
13    public GameObject lockedDoorSensor;
14    public lv1tutorial1To2 thisScript;
15
16    public float timer = 5;
17    private bool doneWalktutorial = true;
18    private bool doneTutorial = false;
19    private float inputX;
20    private float inputY;
21
22    private void Start()
23    {
24        anim = GetComponent<Animator>();
25    }
26
27    // Update is called once per frame
28    void Update()
29    {
30        //moving left right
31        inputX = Input.GetAxis("Horizontal");
32
33        rb.velocity = new Vector2(inputX * walkspeed, rb.velocity.y);
34
35        if (inputX > 0.01f)
36
37            rb.velocity = new Vector2(inputX * walkspeed, rb.velocity.y);
38
39        if (inputX > 0.01f)
40        {
41            transform.localScale = new Vector2(0.15f, 0.15f);
42        }
43        else if (inputX < -0.01f)
44        {
45            transform.localScale = new Vector2(-0.15f, 0.15f);
46        }
47
48        if (doneWalktutorial)
49        {
50            if (inputX != 0)
51            {
52                doneTutorial = true;
53            }
54
55            if (doneTutorial)
56            {
57                timer -= Time.deltaTime;
58                if (timer <= 0)
59                {
60                    walktutorial.SetActive(false);
61                    tutorial2.SetActive(true);
62                    lockedDoorSensor.SetActive(true);
63                    thisScript.enabled = false;
64                    doneWalktutorial = false;
65                }
66            }
67
68            anim.SetBool("IsWalk", inputX != 0);
69        }
70    }
71
72    }
73
74    }
75
76    }
77
78    }
79
80    }
81
82    }
83
84    }
85
86    }
87
88    }
89
90    }
91
92    }
93
94    }
95
96    }
97
98    }
99
100   }
101   }
102   }
103   }
104   }
105   }
106   }
107   }
108   }
109   }
110   }
111   }
112   }
113   }
114   }
115   }
116   }
117   }
118   }
119   }
120   }
121   }
122   }
123   }
124   }
125   }
126   }
127   }
128   }
129   }
130   }
131   }
132   }
133   }
134   }
135   }
136   }
137   }
138   }
139   }
140   }
141   }
142   }
143   }
144   }
145   }
146   }
147   }
148   }
149   }
150   }
151   }
152   }
153   }
154   }
155   }
156   }
157   }
158   }
159   }
160   }
161   }
162   }
163   }
164   }
165   }
166   }
167   }
168   }
169   }
170   }
171   }
172   }
173   }
174   }
175   }
176   }
177   }
178   }
179   }
180   }
181   }
182   }
183   }
184   }
185   }
186   }
187   }
188   }
189   }
190   }
191   }
192   }
193   }
194   }
195   }
196   }
197   }
198   }
199   }
200   }
201   }
202   }
203   }
204   }
205   }
206   }
207   }
208   }
209   }
210   }
211   }
212   }
213   }
214   }
215   }
216   }
217   }
218   }
219   }
220   }
221   }
222   }
223   }
224   }
225   }
226   }
227   }
228   }
229   }
230   }
231   }
232   }
233   }
234   }
235   }
236   }
237   }
238   }
239   }
240   }
241   }
242   }
243   }
244   }
245   }
246   }
247   }
248   }
249   }
250   }
251   }
252   }
253   }
254   }
255   }
256   }
257   }
258   }
259   }
260   }
261   }
262   }
263   }
264   }
265   }
266   }
267   }
268   }
269   }
270   }
271   }
272   }
273   }
274   }
275   }
276   }
277   }
278   }
279   }
280   }
281   }
282   }
283   }
284   }
285   }
286   }
287   }
288   }
289   }
290   }
291   }
292   }
293   }
294   }
295   }
296   }
297   }
298   }
299   }
300   }
301   }
302   }
303   }
304   }
305   }
306   }
307   }
308   }
309   }
310   }
311   }
312   }
313   }
314   }
315   }
316   }
317   }
318   }
319   }
320   }
321   }
322   }
323   }
324   }
325   }
326   }
327   }
328   }
329   }
330   }
331   }
332   }
333   }
334   }
335   }
336   }
337   }
338   }
339   }
340   }
341   }
342   }
343   }
344   }
345   }
346   }
347   }
348   }
349   }
350   }
351   }
352   }
353   }
354   }
355   }
356   }
357   }
358   }
359   }
360   }
361   }
362   }
363   }
364   }
365   }
366   }
367   }
368   }
369   }
370   }
371   }
372   }
373   }
374   }
375   }
376   }
377   }
378   }
379   }
380   }
381   }
382   }
383   }
384   }
385   }
386   }
387   }
388   }
389   }
390   }
391   }
392   }
393   }
394   }
395   }
396   }
397   }
398   }
399   }
400   }
401   }
402   }
403   }
404   }
405   }
406   }
407   }
408   }
409   }
410   }
411   }
412   }
413   }
414   }
415   }
416   }
417   }
418   }
419   }
420   }
421   }
422   }
423   }
424   }
425   }
426   }
427   }
428   }
429   }
430   }
431   }
432   }
433   }
434   }
435   }
436   }
437   }
438   }
439   }
440   }
441   }
442   }
443   }
444   }
445   }
446   }
447   }
448   }
449   }
450   }
451   }
452   }
453   }
454   }
455   }
456   }
457   }
458   }
459   }
460   }
461   }
462   }
463   }
464   }
465   }
466   }
467   }
468   }
469   }
470   }
471   }
472   }
473   }
474   }
475   }
476   }
477   }
478   }
479   }
480   }
481   }
482   }
483   }
484   }
485   }
486   }
487   }
488   }
489   }
490   }
491   }
492   }
493   }
494   }
495   }
496   }
497   }
498   }
499   }
500   }
501   }
502   }
503   }
504   }
505   }
506   }
507   }
508   }
509   }
510   }
511   }
512   }
513   }
514   }
515   }
516   }
517   }
518   }
519   }
520   }
521   }
522   }
523   }
524   }
525   }
526   }
527   }
528   }
529   }
530   }
531   }
532   }
533   }
534   }
535   }
536   }
537   }
538   }
539   }
540   }
541   }
542   }
543   }
544   }
545   }
546   }
547   }
548   }
549   }
550   }
551   }
552   }
553   }
554   }
555   }
556   }
557   }
558   }
559   }
560   }
561   }
562   }
563   }
564   }
565   }
566   }
567   }
568   }
569   }
570   }
571   }
572   }
573   }
574   }
575   }
576   }
577   }
578   }
579   }
580   }
581   }
582   }
583   }
584   }
585   }
586   }
587   }
588   }
589   }
590   }
591   }
592   }
593   }
594   }
595   }
596   }
597   }
598   }
599   }
600   }
601   }
602   }
603   }
604   }
605   }
606   }
607   }
608   }
609   }
610   }
611   }
612   }
613   }
614   }
615   }
616   }
617   }
618   }
619   }
620   }
621   }
622   }
623   }
624   }
625   }
626   }
627   }
628   }
629   }
630   }
631   }
632   }
633   }
634   }
635   }
636   }
637   }
638   }
639   }
640   }
641   }
642   }
643   }
644   }
645   }
646   }
647   }
648   }
649   }
650   }
651   }
652   }
653   }
654   }
655   }
656   }
657   }
658   }
659   }
660   }
661   }
662   }
663   }
664   }
665   }
666   }
667   }
668   }
669   }
670   }
671   }
672   }
673   }
674   }
675   }
676   }
677   }
678   }
679   }
680   }
681   }
682   }
683   }
684   }
685   }
686   }
687   }
688   }
689   }
690   }
691   }
692   }
693   }
694   }
695   }
696   }
697   }
698   }
699   }
700   }
701   }
702   }
703   }
704   }
705   }
706   }
707   }
708   }
709   }
710   }
711   }
712   }
713   }
714   }
715   }
716   }
717   }
718   }
719   }
720   }
721   }
722   }
723   }
724   }
725   }
726   }
727   }
728   }
729   }
730   }
731   }
732   }
733   }
734   }
735   }
736   }
737   }
738   }
739   }
740   }
741   }
742   }
743   }
744   }
745   }
746   }
747   }
748   }
749   }
750   }
751   }
752   }
753   }
754   }
755   }
756   }
757   }
758   }
759   }
760   }
761   }
762   }
763   }
764   }
765   }
766   }
767   }
768   }
769   }
770   }
771   }
772   }
773   }
774   }
775   }
776   }
777   }
778   }
779   }
780   }
781   }
782   }
783   }
784   }
785   }
786   }
787   }
788   }
789   }
790   }
791   }
792   }
793   }
794   }
795   }
796   }
797   }
798   }
799   }
800   }
801   }
802   }
803   }
804   }
805   }
806   }
807   }
808   }
809   }
810   }
811   }
812   }
813   }
814   }
815   }
816   }
817   }
818   }
819   }
820   }
821   }
822   }
823   }
824   }
825   }
826   }
827   }
828   }
829   }
830   }
831   }
832   }
833   }
834   }
835   }
836   }
837   }
838   }
839   }
840   }
841   }
842   }
843   }
844   }
845   }
846   }
847   }
848   }
849   }
850   }
851   }
852   }
853   }
854   }
855   }
856   }
857   }
858   }
859   }
860   }
861   }
862   }
863   }
864   }
865   }
866   }
867   }
868   }
869   }
870   }
871   }
872   }
873   }
874   }
875   }
876   }
877   }
878   }
879   }
880   }
881   }
882   }
883   }
884   }
885   }
886   }
887   }
888   }
889   }
890   }
891   }
892   }
893   }
894   }
895   }
896   }
897   }
898   }
899   }
900   }
901   }
902   }
903   }
904   }
905   }
906   }
907   }
908   }
909   }
910   }
911   }
912   }
913   }
914   }
915   }
916   }
917   }
918   }
919   }
920   }
921   }
922   }
923   }
924   }
925   }
926   }
927   }
928   }
929   }
930   }
931   }
932   }
933   }
934   }
935   }
936   }
937   }
938   }
939   }
940   }
941   }
942   }
943   }
944   }
945   }
946   }
947   }
948   }
949   }
950   }
951   }
952   }
953   }
954   }
955   }
956   }
957   }
958   }
959   }
960   }
961   }
962   }
963   }
964   }
965   }
966   }
967   }
968   }
969   }
970   }
971   }
972   }
973   }
974   }
975   }
976   }
977   }
978   }
979   }
980   }
981   }
982   }
983   }
984   }
985   }
986   }
987   }
988   }
989   }
990   }
991   }
992   }
993   }
994   }
995   }
996   }
997   }
998   }
999   }
1000  }

```

Figure 4.34 lv1tutorial1To2 Script in Tutorial 1

Figure 4.35 shows the Interface of the Tutorial 2. The tutorial is to teach players how to open the door by pressing E button.



Figure 4.35 Interface of the Tutorial 2 In Level 1

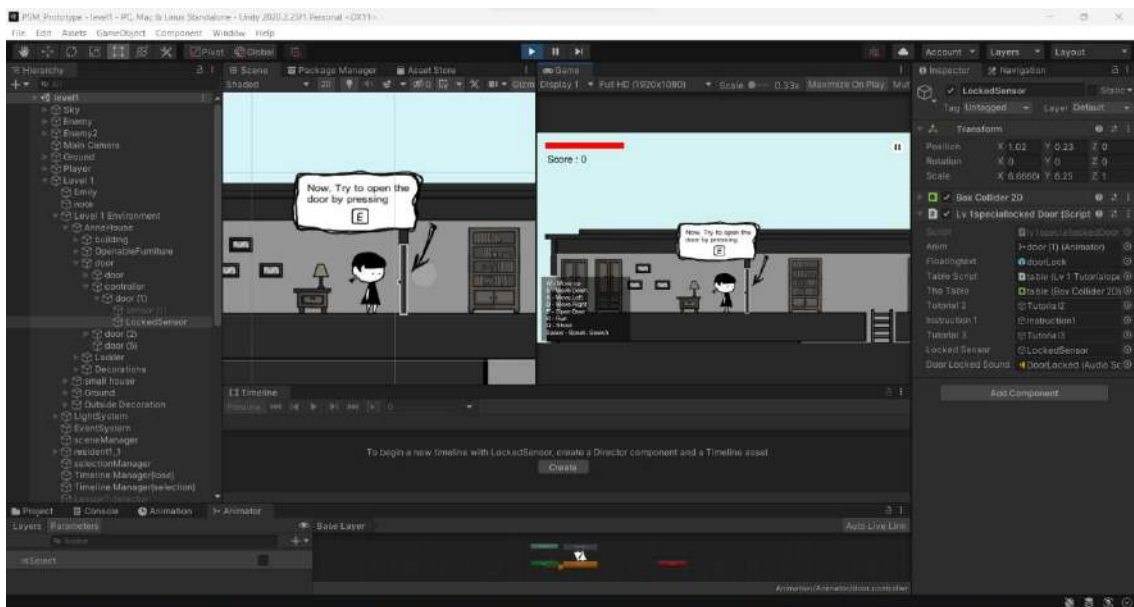


Figure 4.36 Tutorial 2 Open Door Development

Figure 4.37 shows the `lv1speciallockedDoor` Script of the Tutorial 2 in Level 1. When players touch the door and press E button, the *float timer* will start to count down for 5 seconds. After 5 seconds, the Tutorial 2 panel will be hidden and the Tutorial 3 will be shown.

```

lv1speciallockedDoor.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class lv1speciallockedDoor : MonoBehaviour
6
7 {
8     public Animator anim;
9     private bool TouchDoor = false;
10    public GameObject floatingText;
11    private float timer = 5f;
12    public lv1TutorialOpenFurniture tableScript;
13    public BoxCollider2D theTable;
14    public GameObject tutorial1;
15    public GameObject tutorial2;
16    public GameObject tutorial3;
17    public GameObject lockedSensor;
18    private bool showInst1 = false;
19
20    public AudioSource doorLockedSound;
21
22    // Start is called before the first frame update
23    void Start()
24    {
25
26    }
27
28    void Update()
29    {
30        if (TouchDoor && Input.GetKeyDown("e"))
31        {
32            doorLockedSound.Play();
33            Instantiate(floatingText, transform.position, Quaternion.identity);
34
35            if (TouchDoor && Input.GetKeyDown("e"))
36            {
37                SHOWInst1 = true;
38
39            }
40
41            if (showInst1)
42
43            {
44                tutorial2.SetActive(false);
45                instruction1.SetActive(true);
46
47                timer -= Time.deltaTime;
48                if(timer <= 0)
49                {
50                    instruction1.SetActive(false);
51                    tutorial3.SetActive(true);
52                    theTable.enabled = true;
53                    tableScript.enabled = true;
54                    showInst1 = false;
55                    lockedSensor.SetActive(false);
56                }
57            }
58
59            private void OnTriggerEnter2D(Collider2D collision)
60            {
61                if (collision.CompareTag("Player"))
62                {
63                    anim.SetBool("isSelect", true);
64                    TouchDoor = true;
65                }
66            }
67
68            private void OnTriggerExit2D(Collider2D collision)
69            {
70                if (collision.CompareTag("Player"))
71                {
72                    anim.SetBool("isSelect", false);
73                    TouchDoor = false;
74                }
75            }

```

Figure 4.37 lv1speciallockedDoor Script in Tutorial 2

Figure 4.38 shows the Interface of the Tutorial 3 in Level 1. The tutorial is to teach players how to find key and search for important message in the furniture by pressing SPACE.

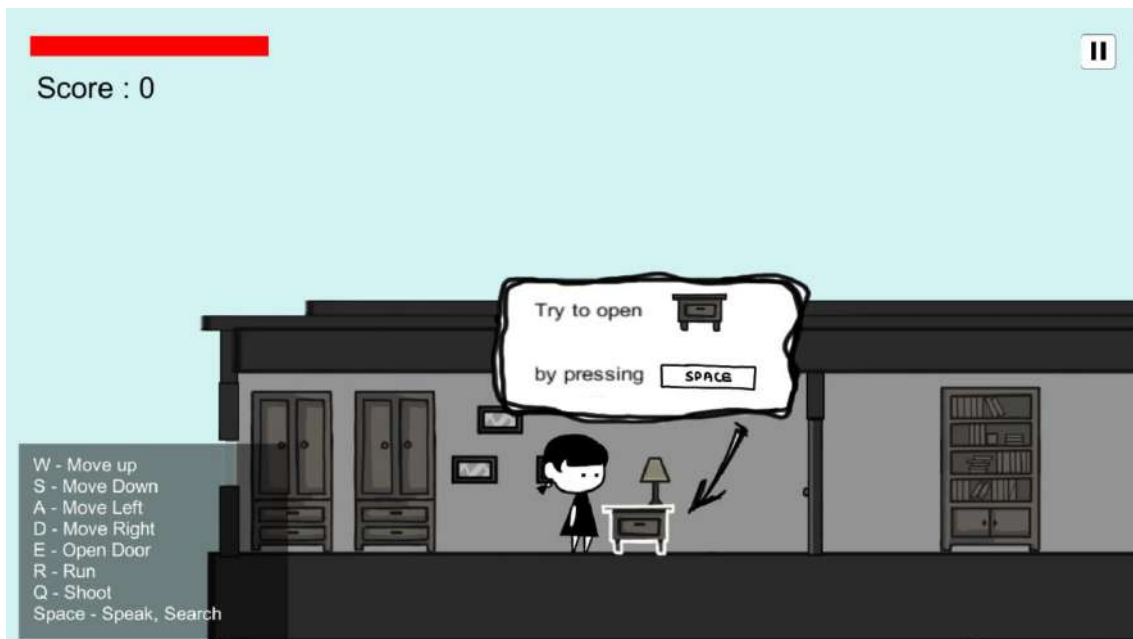


Figure 4.38 Interface of the Tutorial 3 in Level 1

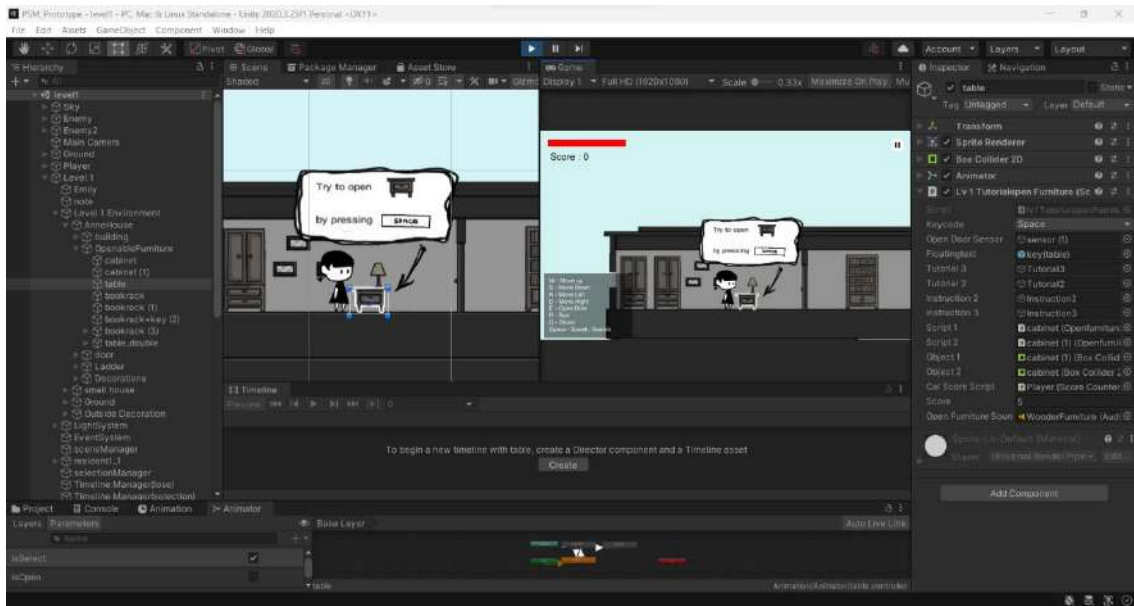


Figure 4.39 Tutorial 3 Open Furniture Development

Figure 4.40 shows the `lv1TutorialopenFurniture` Script of the Tutorial 3 in Level 1. When players touched the table and press SPACE, the Tutorial 3 panel will be hidden and the Tutorial 4 will be shown.

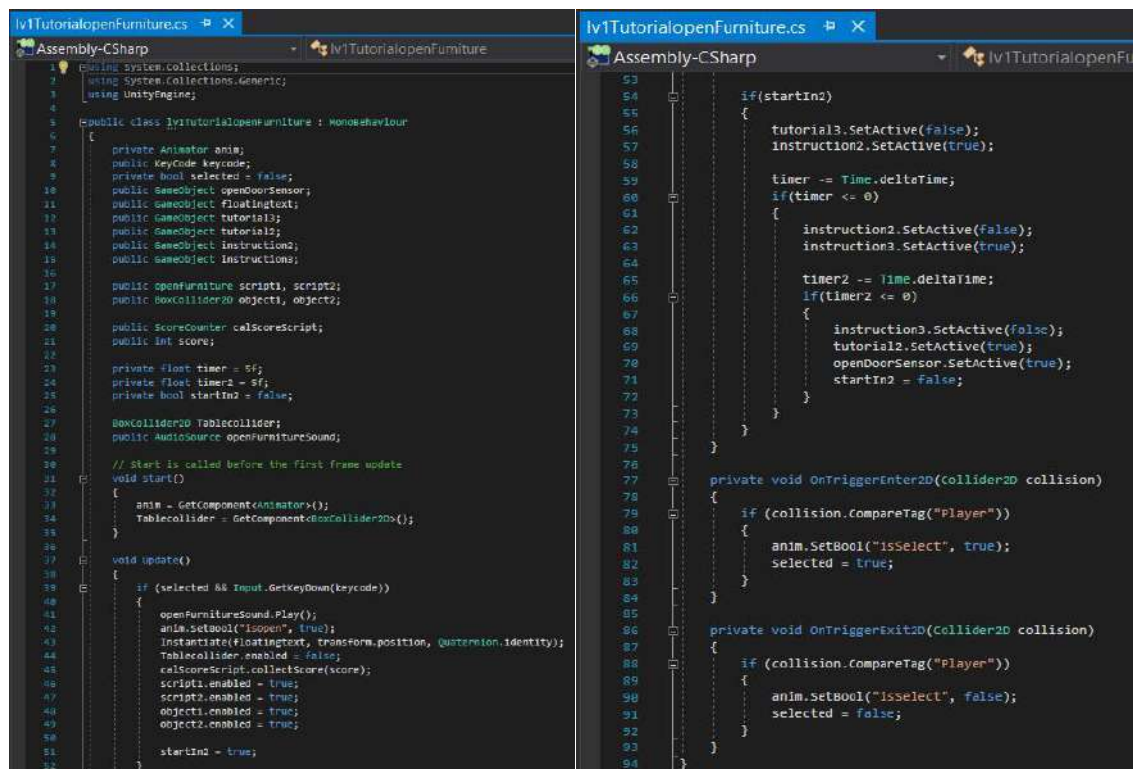


Figure 4.40 `lv1TutorialopenFurniture` Script in Tutorial 3

Figure 4.41 shows the Interface of the Read Message / Note Tutorial in Level 1. The tutorial teach players how to read the message / note by pressing SPACE.



Figure 4.41 Interface of the Read Message / Note Tutorial in Level 1

Figure 4.42 shows the Message that leave by Anne's Mom. This is a important message to teach players do not open the door for a stranger when they are alone in home.

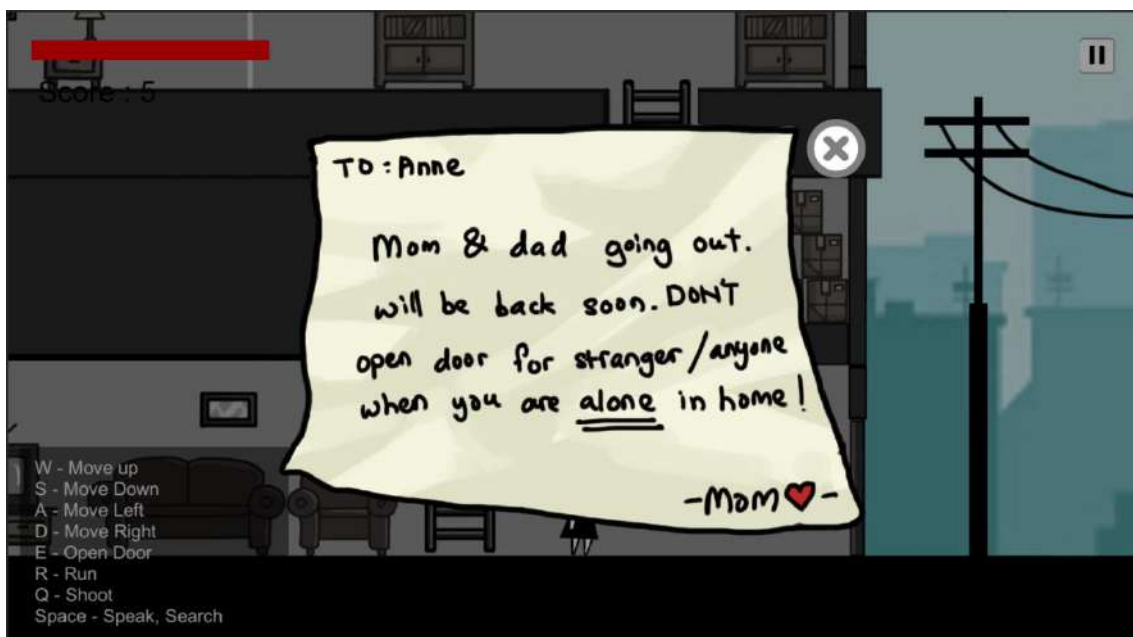


Figure 4.42 Message Leave by Anne's Mom in Level 1



Figure 4.43 Clue 1 from Emily Interface in Level 1

Figure 4.44 shows the development of the Clue 1 from Emily in Level 1. The collider is placed on the Double Table and when players collide and press SPACE, the Clue 1 panel will be shown.

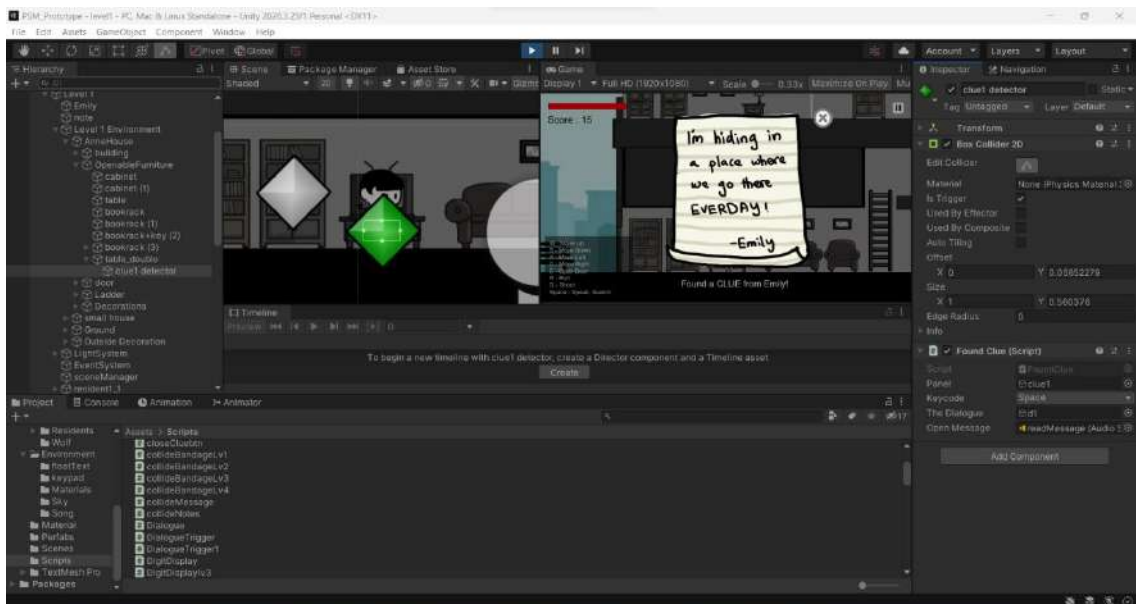


Figure 4.44 Clue 1 from Emily Development in Level 1



Figure 4.45 Poster About “Don’t Open Doors To Stranger” Interface in Level 1

Figure 4.46 shows the development of the Poster About “Don’t Open Doors To Stranger” in Level 1. The collider is placed on a book rack and when players collide and press SPACE, the Poster panel will be shown.

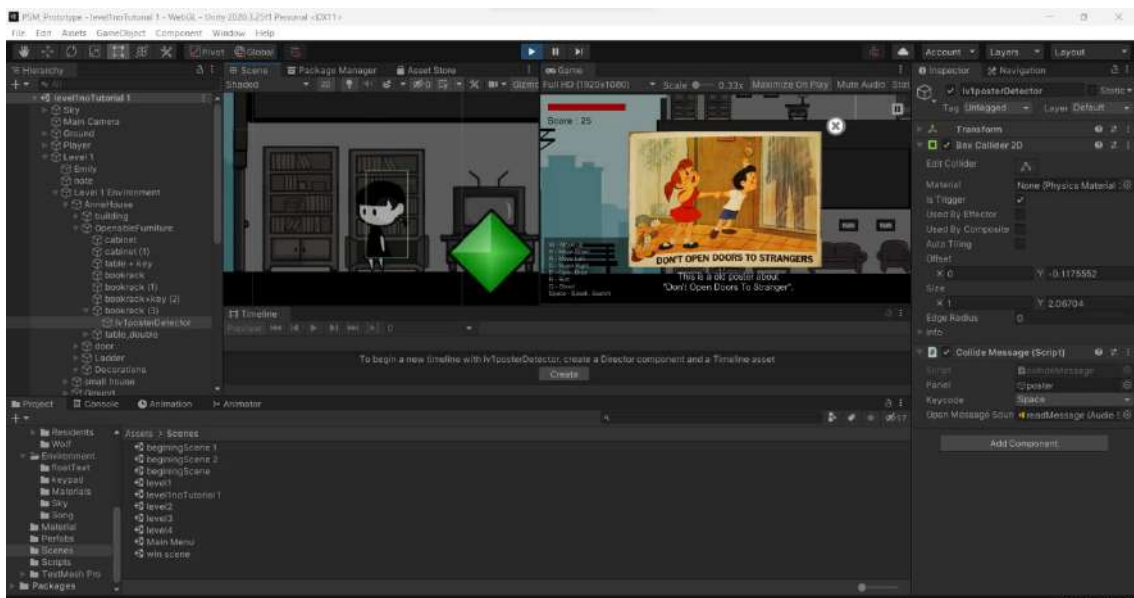


Figure 4.46 Poster About “Don’t Open Doors To Stranger” Development in Level 1

Figure 4.47 shows the Level 1 Lesson Timeline. The timeline will play the animation about a stranger try to ask Anne to open the door for him by lying that Anne's mother ask him to take care of her.



Figure 4.47 Level 1 Lesson Timeline Interface

Figure 4.48 shows the Lesson 1 development. The lesson detector (2 in the figure) will be shown after players open the book rack (1 in the figure). The lesson detector is used to play the Timeline Lesson.

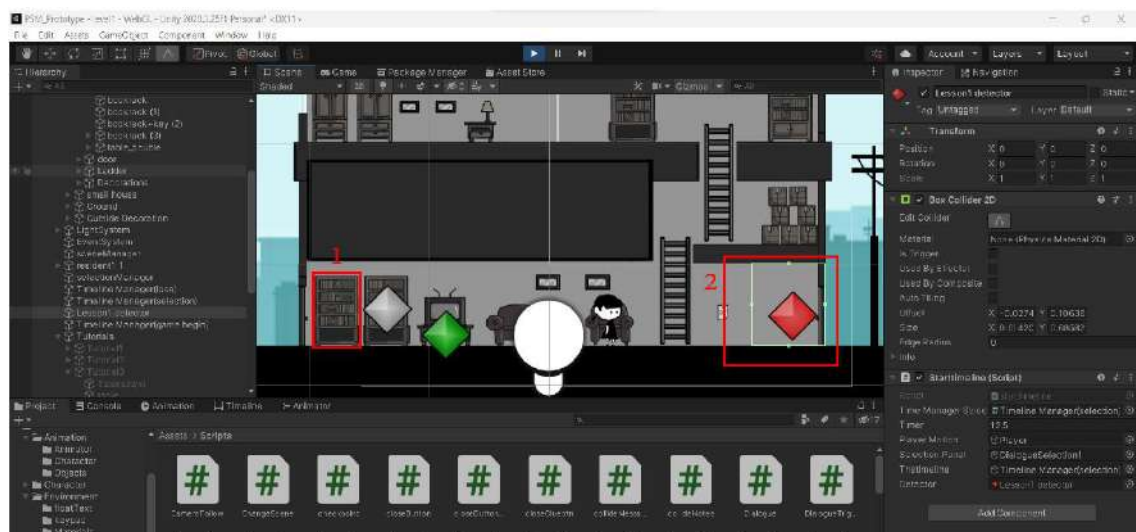


Figure 4.48 Level 1 Lesson Timeline Interface Development

Figure 4.49 shows the starttimeline Script of the Lesson Detector in Level 1. When the player collide the Lesson Detector, the Lesson timeline will be played. The timer will start count down for 9 seconds. After 9 seconds, the Level 1 Selection Panel will be shown.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.Playables;
6
7 public class starttimeline : MonoBehaviour
8 {
9     public PlayableDirector TimeManagerSelectionLV1;
10    public float timer;
11    public GameObject playerMotion;
12    public GameObject SelectionPanel;
13    public GameObject theTimeline;
14    public GameObject detector;
15
16    private bool isPlaying = false;
17
18    void OnCollisionEnter2D(Collision2D col)
19    {
20        if(col.gameObject.CompareTag("Player"))
21        {
22            playerMotion.GetComponent<Animator>().SetBool("ismwalk", false);
23            TimeManagerSelectionLV1.Play();
24            isPlaying = true;
25        }
26    }
27
28
29    private void Update()
30    {
31        if(isPlaying)
32        {
33            timer -= Time.deltaTime;
34            if (timer <= 0)
35            {
36                Time.timeScale = 0;
37                detector.GetComponent<BoxCollider2D>().enabled = false;
38                SelectionPanel.SetActive(true);
39                isPlaying = false;
40                timer = 9f;
41            }
42        }
43    }
44
45
46

```

Figure 4.49 starttimeline Script in Lesson Detector Level 1

Figure 4.50 shows the Level 1 Selection Panel Interface. The selection panel show a question to player if want to open the door for a stranger or not. There are 2 buttons provided which is Yes and No.

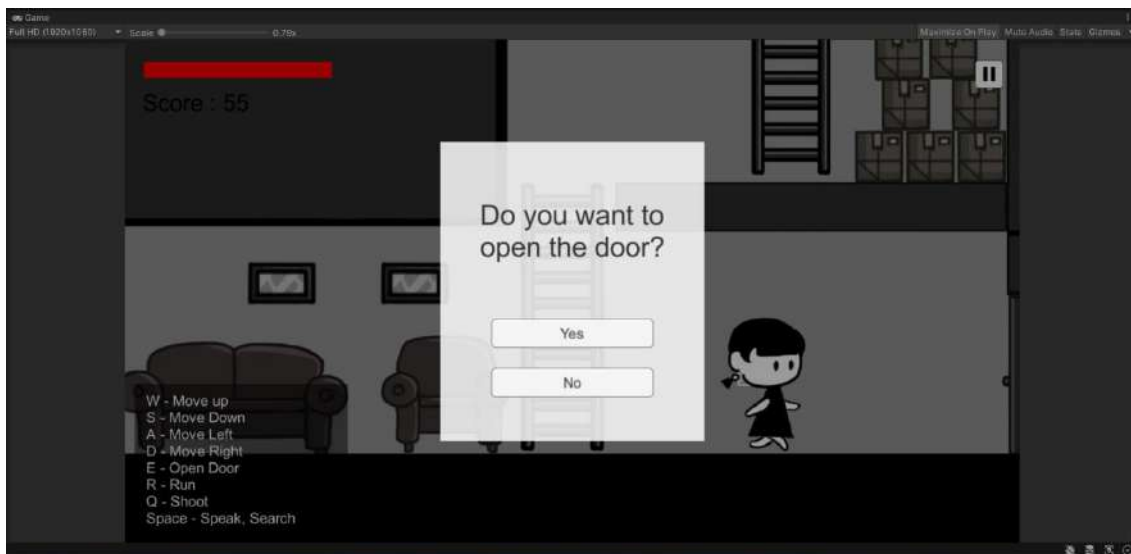


Figure 4.50 Level 1 Selection Panel Interface

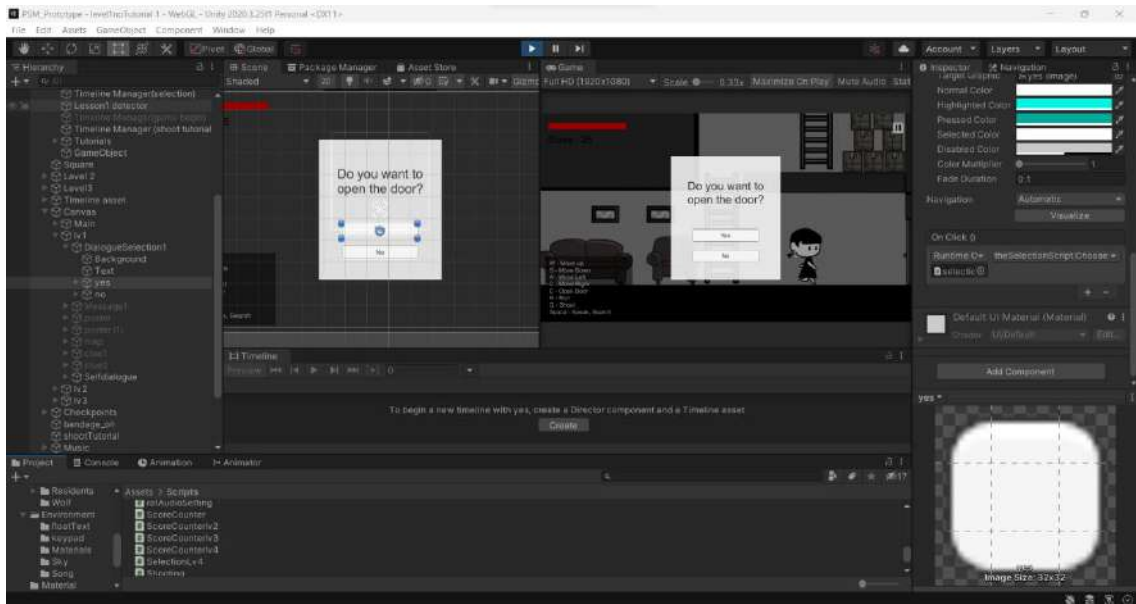


Figure 4.51 Level 1 Selection Panel Interface Development

Figure 4.52 shows the theSelectionScript Script for Selection Panel in Level 1. When players click the Yes button in the Selection Panel, the script will run the *public void ChooseYes()*, then the lose timeline will be played, while players click the No button on the Selection panel, the script will run the *public void ChooseNo()*, the Selection Panel will be hidden and players are able to continue the game.

```

theSelectionScript.cs
Assembly-CSharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Playables;
using UnityEngine.UI;

public class theSelectionScript : MonoBehaviour
{
    public GameObject selection;
    public GameObject playermove;
    public timelineManager script;
    public Camera cam;
    public bool lose = false;

    public AudioSource walksound;

    public void chooseyes()
    {
        selection.SetActive(false);
        Time.timeScale = 1;
        lose = true;
        script.timelineState(lose);
    }

    public void chooseno()
    {
        selection.SetActive(false);
        Time.timeScale = 1;
        playermove.GetComponent<PlayerController>().enabled = true;
        cam.GetComponent<CameraFollow>().enabled = true;
        cam.transform.position = new Vector3(0, 0, -10);
        cam.orthographicSize = 1.15f;
        walksound.volume = 1f;
    }
}

```

Figure 4.52 theSelectionScript Script for Selection Panel in Level 1

Figure 4.53 shows the Shooting Tutorial Timeline Interface in Level 1. The tutorial timeline played the animation and teach players to shoot to defeat the enemy by pressing Q button.



Figure 4.53 Shooting Tutorial Timeline Interface in Level 1

Figure 4.54 shows the Shooting Tutorial Timeline Development in Level 1. When players collide with thw collider, the Shooting Tutorial Timeline will be played. After finish the tutorial, players are able to shoot by pressing Q button to defeat the enemy.

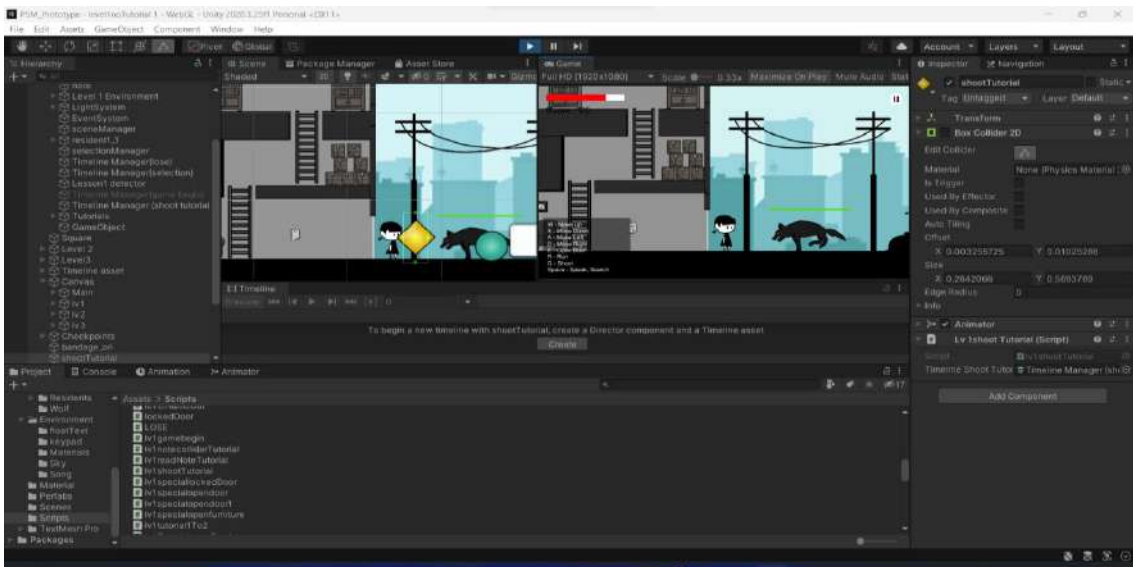


Figure 4.54 Shooting Tutorial Timeline Development in Level 1

Figure 4.55 shows the Resident Dialogue in Level 1. Players able to know where Emily goes by talking with this resident.



Figure 4.55 Resident Dialogue in Level 1

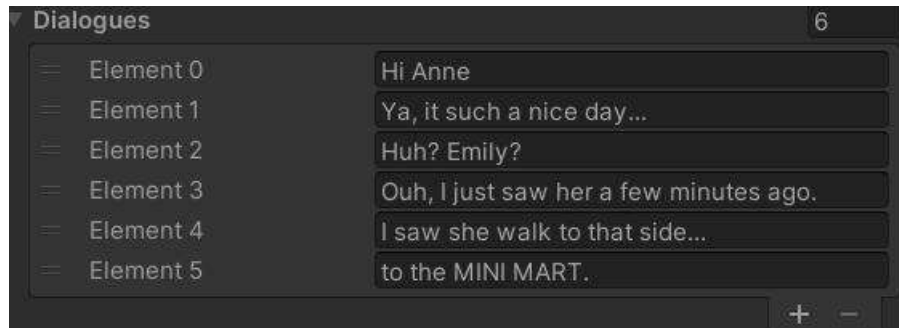


Figure 4.56 Dialogue of the Resident in Level 1

Figure 4.57 shows the Map drawn by Emily in Level 1. Players able to view the Levels of the game in the map. There are 4 levels which are Anne's House, Mini Mart, Emily's House and School.

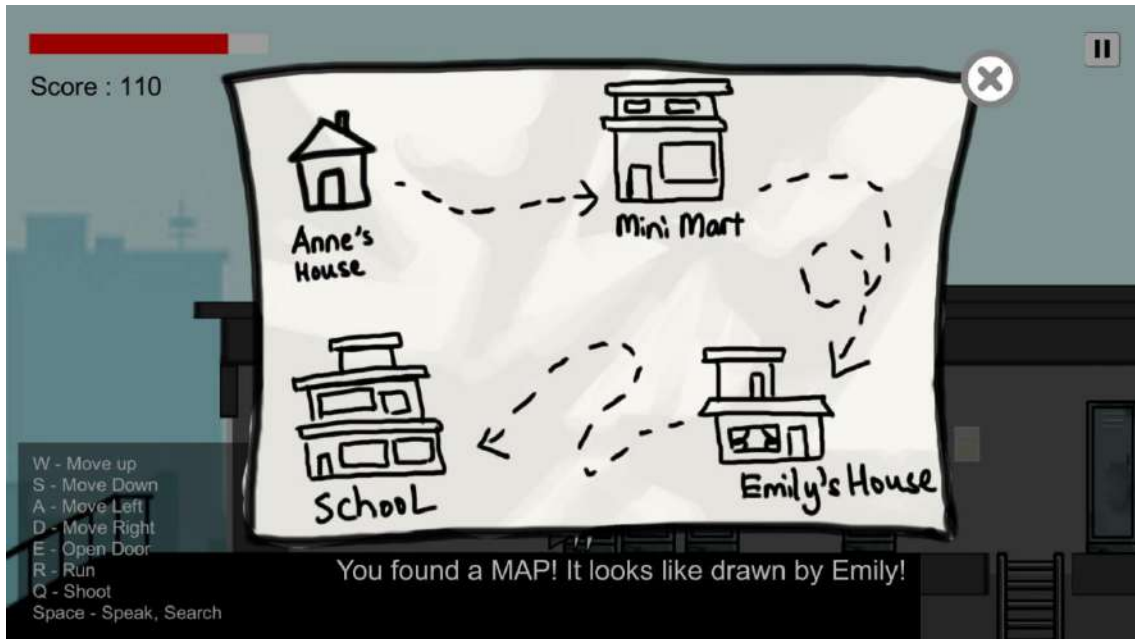


Figure 4.57 Map Drawn by Emily in Level 1

Figure 4.58 shows the development of the Map in level 1. When players collide with the collider and press SPACE, the Map Panel will be shown.

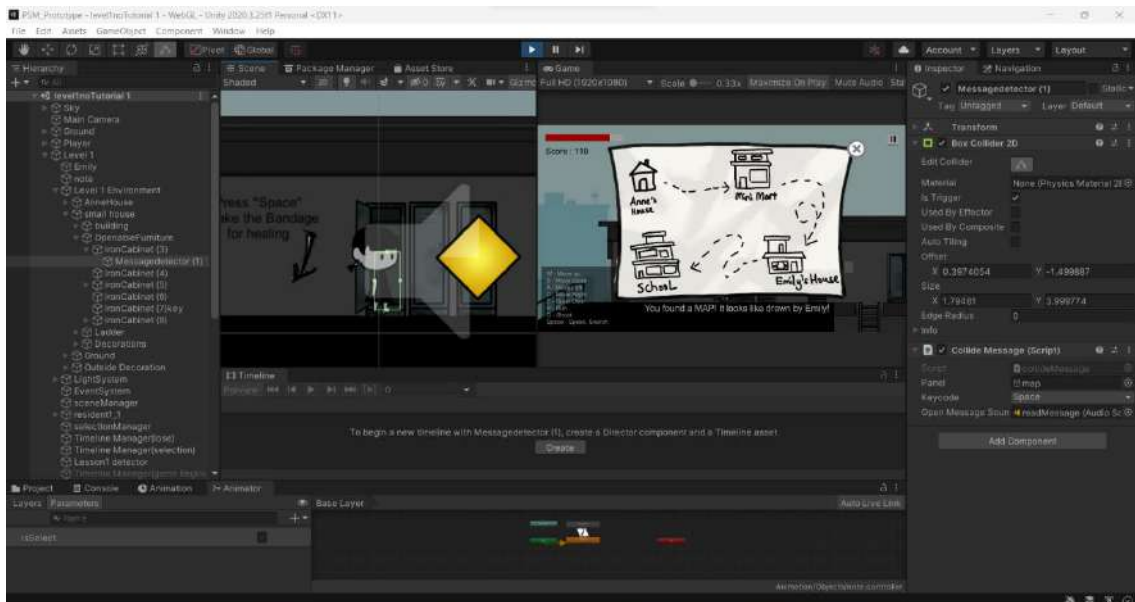


Figure 4.58 Map Drawn by Emily Interface Development in Level 1

Figure 4.59 shows the “Kidnapping Cases In Malaysia” News Interface in Level 1. The news is about the kidnapping cases in Malaysia between 2020 and the early 2022. The

purpose of the news apply in the game is to increase the awareness of kidnapping in players



Figure 4.59 “Kidnapping Cases In Malaysia” News Interface in Level 1

Figure 4.60 shows the “Kidnapping Cases In Malaysia” News Interface Development in Level 1. When player collide with the collider and press SPACE, the news panel will be shown.

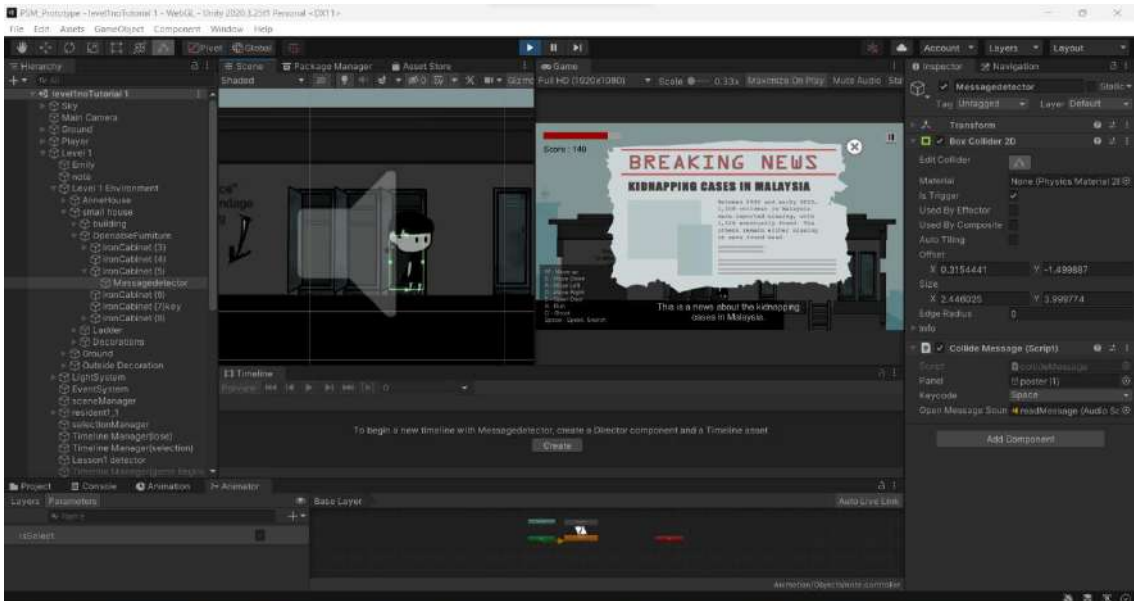


Figure 4.60 “Kidnapping Cases In Malaysia” News Interface Development in Level 1

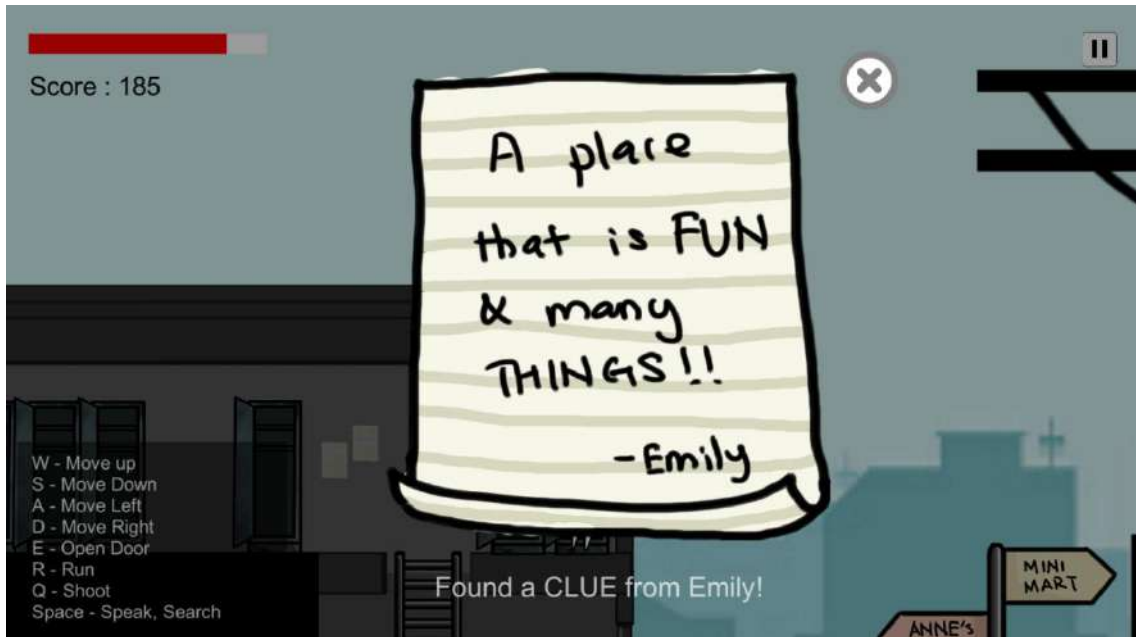


Figure 4.61 Clue 2 From Emily in Level 1

Figure 4.62 shows the Checkpoint development in Level 1. When players collide the checkpoint collide, the script will load the scene to Level 2.

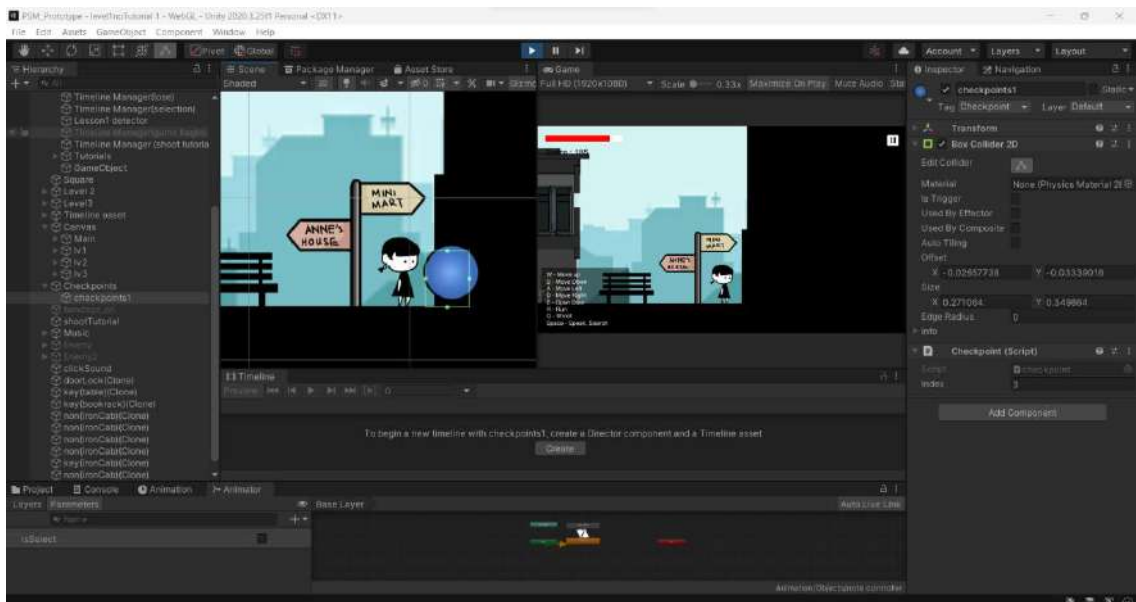


Figure 4.62 Checkpoint Development in Level 1

Level 2- Mini Mart



Figure 4.63 Game Environment Design of Level 2

Figure 4.64 shows the Clue 3 from Emily in Level 2. This clue will be shown when players open the table in a small building in Level 2.



Figure 4.64 Clue 3 From Emily in Level 2

Figure 4.65 shows the Resident 1 Dialogue in Level 2. The resident 1 tell the players that his mother said do not walk in back alley.



Figure 4.65 Resident 1 Dialogue in Level 2

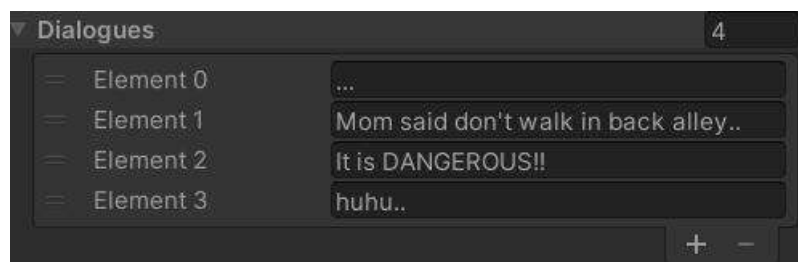


Figure 4.66 Dialogue of the Resident 1 in Level 2

Figure 4.67 shows the Resident 2 Dialogue in Level 2. The resident 2 tell the players that he saw a stranger walking around in the back alley. He ask Anne to be careful and ask help from him if molested by the stranger.



Figure 4.67 Resident 2 Dialogue in Level 2

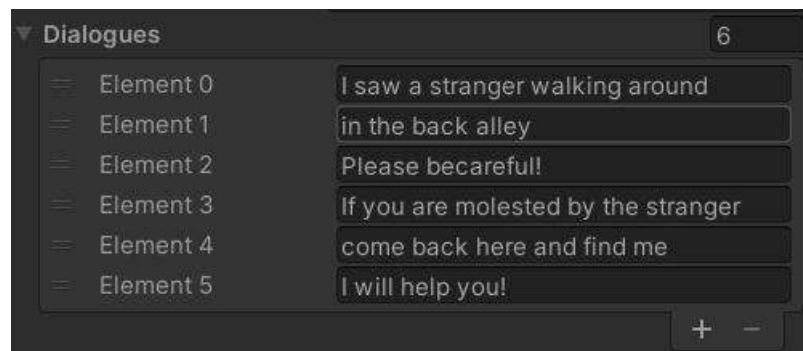


Figure 4.68 Dialogue of the Resident 2 in Level 2

Figure 4.69 shows the Resident 3 (Worker 1) Dialogue in Level 2. The dialogue is important to know the location of the key for locked doors. The resident will tell Anne that the key for locked ground floor door is located in 2nd Floor and the key for locked 1st Floor is located in the Basement.



Figure 4.69 Resident 3 (Worker 1) Dialogue in Level 2

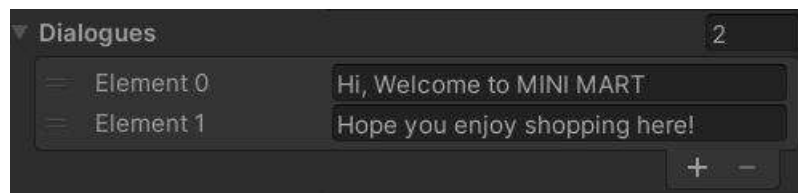


Figure 4.70 First Dialogue of the Resident 3 (Worker 1) in Level 2

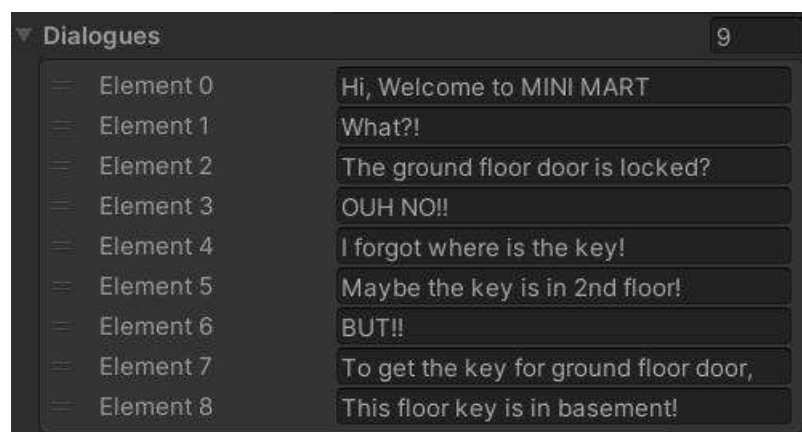


Figure 4.71 Second Dialogue of the Resident 3 (Worker 1) in Level 2

Figure 4.72 shows the Resident 3 (Worker 2) Dialogue in Level 2. The dialogue is important to get the key for locked basement door.



Figure 4.72 Resident 3 (Worker 2) Dialogue in Level 2

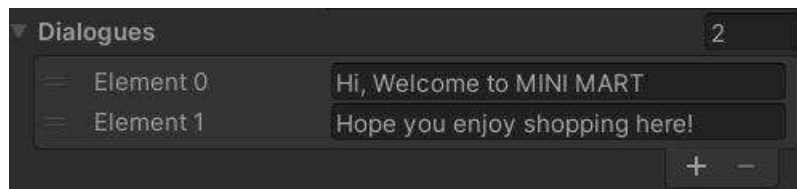


Figure 4.73 First Dialogue of the Resident 3 (Worker 2) in Level 2

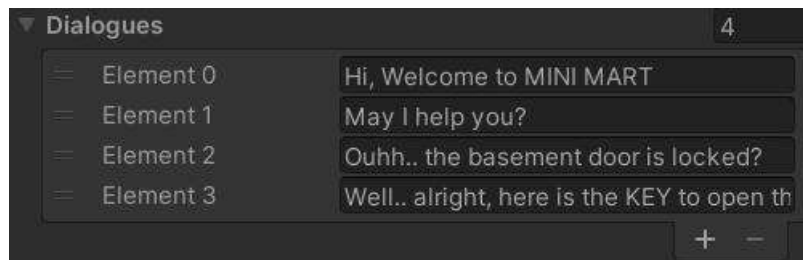


Figure 4.74 Second Dialogue of the Resident 3 (Worker 2) in Level 2

Figure 4.75 shows the Dialogue of Worker 1 & 2 Development in Level 2. The second dialogue of worker 1 & 2 will be activated after players try to open the door on the ground floor. The dialogue is about the worker 1 left the ground floor's key on the second floor. Players have to find the first floor's key in the basement

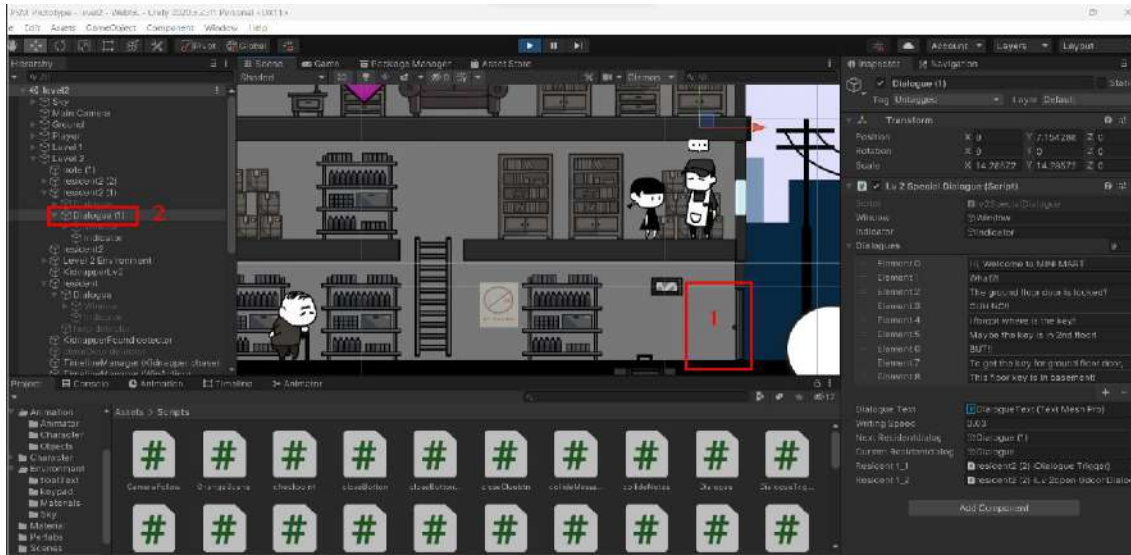


Figure 4.75 Dialogue of Worker 1 & 2 Development in Level 2.

Figure 4.76 shows the Resident 4 Dialogue in Level 2. The dialogue purpose is to increase the awareness of kidnapping in players.

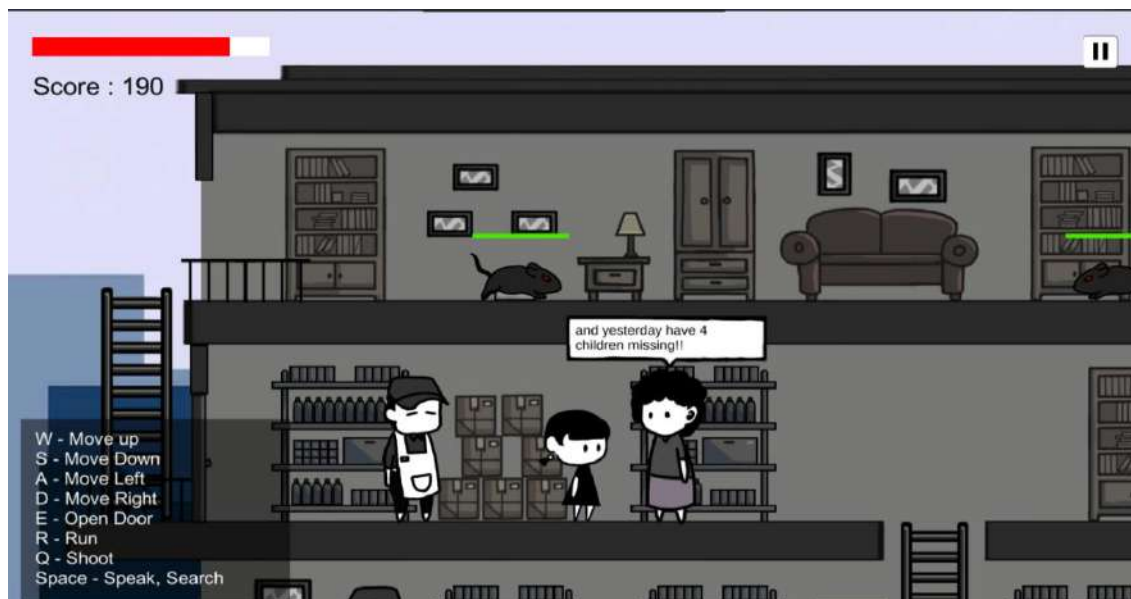


Figure 4.76 Resident 4 Dialogue in Level 2

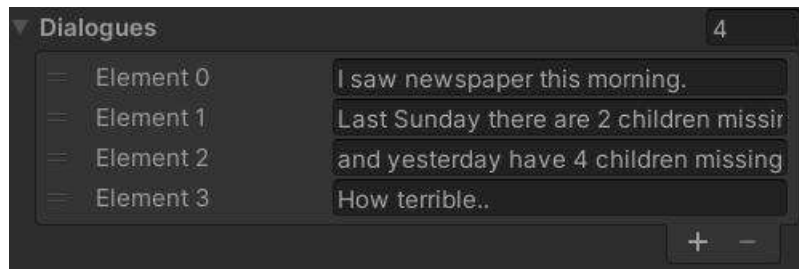


Figure 4.77 Dialogue of the Resident 2 in Level 2

Figure 4.78 shows the Password Note in basement Level 2. The panel shows a code 4608 which will be used in the Locked Door with KeyPad.



Figure 4.78 Keypad Code in Level 2

Figure 4.79 shows the small code hint interface. The small hint will be shown after players click the close button on the Password Note Panel. The hint will be deleted after players successfully open the keypad.



Figure 4.79 Small Code Hint Interface in Level 2

Figure 4.80 shows the KeyPad Panel Interface in Level 2. The door that is locked with a keypad will show the KeyPad Panel when players collide the door and press E. Players are required to find the code to open the KeyPad. Players have to click the 4 correct numbers using the mouse cursor, after that click the green tick button.

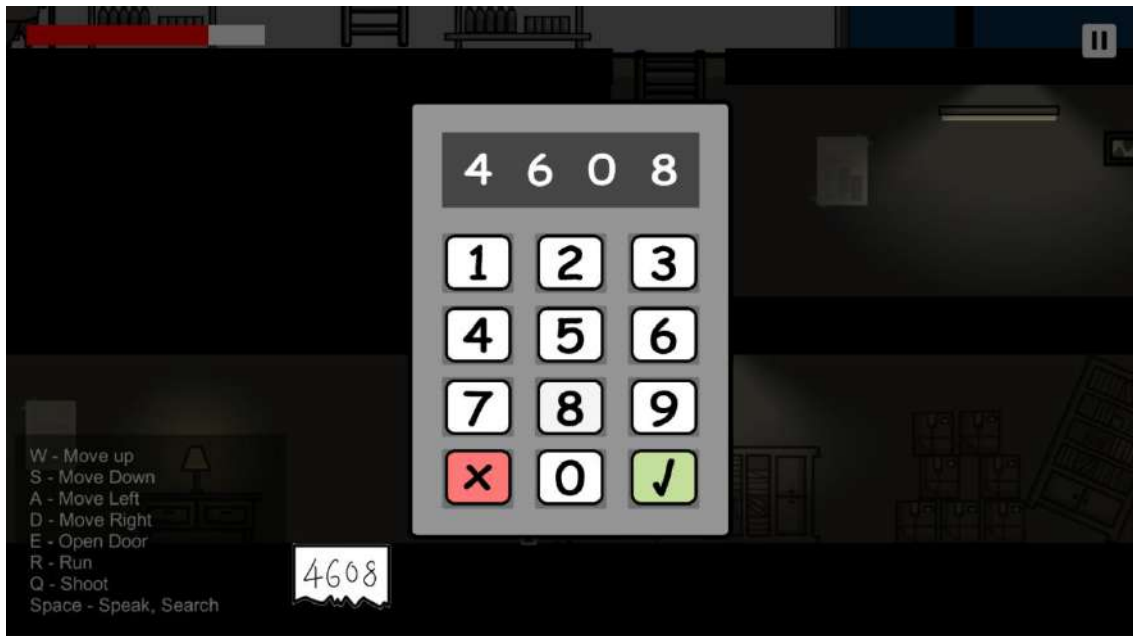


Figure 4.80 KeyPad Panel Interface in Level 2

Figure 4.81 shows the PushTheBtn Script for Keypad Button in Level 2. When players press any button on the Keypad Panel, the script will get the index number of the Number.

```

PushTheBtn.cs | checkpoints
Assembly-CSharp | PushTheBtn
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System;
5 using UnityEngine.UI;
6
7 public class PushTheBtn : MonoBehaviour
8 {
9     public static event Action<string> btnPressed = delegate { };
10
11     private int dividerPosition;
12     private string btnName, btnValue;
13
14     public AudioSource clickSound;
15
16     // Start is called before the first frame update
17     void Start()
18     {
19         btnName = gameObject.name;
20         dividerPosition = btnName.IndexOf(".");
21         btnValue = btnName.Substring(0, dividerPosition);
22         gameObject.GetComponent<Button>().onClick.AddListener(ButtonClicked);
23     }
24
25     private void ButtonClicked()
26     {
27         clickSound.Play();
28         btnPressed(btnValue);
29     }
30
31 }
    
```

Figure 4.81 PushTheBtn Script for Keypad Button in Level 2

Figure 4.82 shows the DigitDisplay Script for Keypad in Level 2. the script is to check the code insert by players is correct or wrong. If correct, the locked door will be hidden else if wrong, the keypad display screen will be reset and players are able to insert correct code.

```

DigitDisplay.cs | DigitDisplay
Assembly-CSharp | DigitDisplay
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using System;
5 using UnityEngine.UI;
6
7 public class DigitDisplay : MonoBehaviour
8 {
9     [SerializeField]
10     private Sprite[] digits;
11
12     [SerializeField]
13     private Image[] characters;
14
15     private string codeSequence;
16     public GameObject keypad;
17     public GameObject tips;
18
19     public AudioSource correctSound;
20     public AudioSource wrongSound;
21
22     // Start is called before the first frame update
23     void Start()
24     {
25         codeSequence = "";
26         for (int i = 0; i < characters.Length - 1; i++)
27         {
28             characters[i].sprite = digits[i];
29         }
30     }
31
32     PushTheBtn btnPressed => AddDigitToCodeSequence();
33
34     private void AddDigitToCodeSequence(string digitEntered)
35     {
36         if (codeSequence.Length < 4)
37         {
38             switch (digitEntered)
39             {
40                 case "0":
41                     codeSequence += "0";
42                     DisplayCodeSequence(0);
43                     break;
44                 case "1":
45                     codeSequence += "1";
46                     DisplayCodeSequence(1);
47                     break;
48                 case "2":
49                     codeSequence += "2";
50                     DisplayCodeSequence(2);
51                     break;
52                 case "3":
53                     codeSequence += "3";
54                     DisplayCodeSequence(3);
55                     break;
56                 case "4":
57                     codeSequence += "4";
58                     DisplayCodeSequence(4);
59                     break;
60                 case "5":
61                     codeSequence += "5";
62                     DisplayCodeSequence(5);
63                     break;
64                 case "6":
65                     codeSequence += "6";
66                     DisplayCodeSequence(6);
67                     break;
68                 case "7":
69                     codeSequence += "7";
70                     DisplayCodeSequence(7);
71                     break;
72                 case "8":
73                     codeSequence += "8";
74                     DisplayCodeSequence(8);
75                     break;
76                 case "9":
77                     codeSequence += "9";
78                     DisplayCodeSequence(9);
79                     break;
80             }
81         }
82     }
83
84     void CheckResult()
85     {
86         if (codeSequence == "6666")
87         {
88             correctSound.Play();
89             Debug.Log("Correct!");
90             keypad.SetActive(false);
91             Destroy(knob);
92             Destroy(tips);
93         }
94         else
95         {
96             wrongSound.Play();
97             Debug.Log("Wrong!");
98             ResetDisplay();
99         }
100     }
101
102     private void ResetDisplay()
103     {
104         for (int i = 0; i < characters.Length - 1; i++)
105         {
106             characters[i].sprite = digits[i];
107         }
108         codeSequence = "";
109     }
110
111     private void OnDestroy()
112     {
113         PushTheBtn btnPressed => AddDigitToCodeSequence();
114     }
115 }
    
```

Figure 4.82 DigitDisplay Script for Keypad in Level 2

Figure 4.83 shows the “Run and Get Help” Message in Level 2. This is a instruction of teaching players run as fast as they could and get help when a kidnapper is approaching.

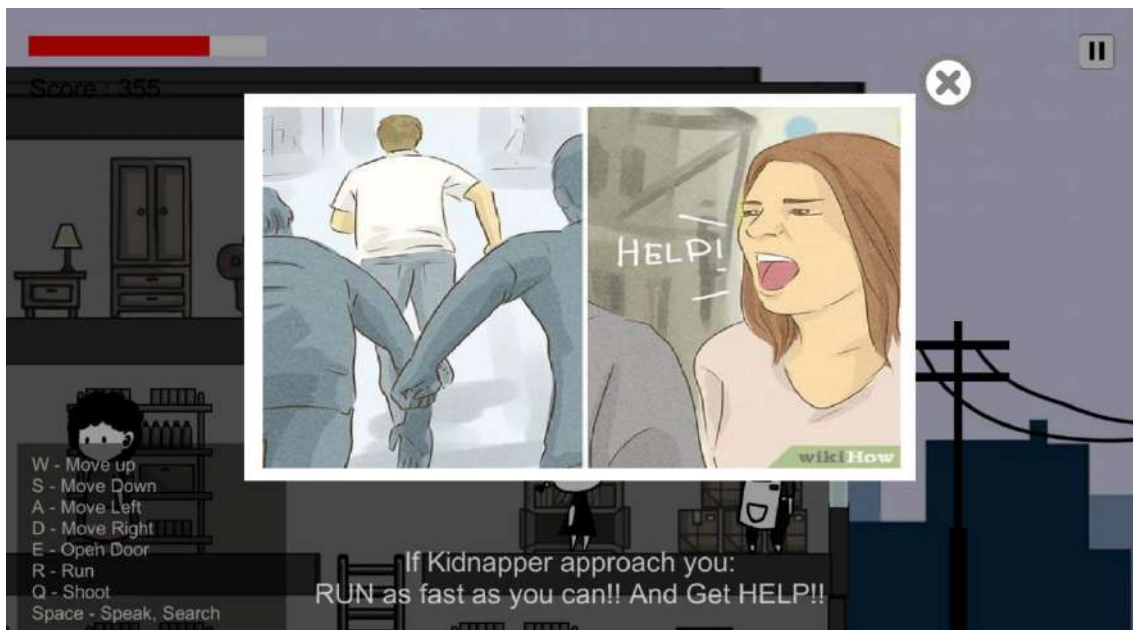


Figure 4.83 “Run and Get Help” Message in Level 2



Figure 4.84 Clue 4 From Emily in Level 2

Figure 4.85 shows the Poster about Do Not Walk in Less Crowded Place in Level 2. The poster teach players that do not walk alone in back alley and avoid less crowded places.



Figure 4.85 Poster about Do Not Walk in Less Crowded Place in Level 2

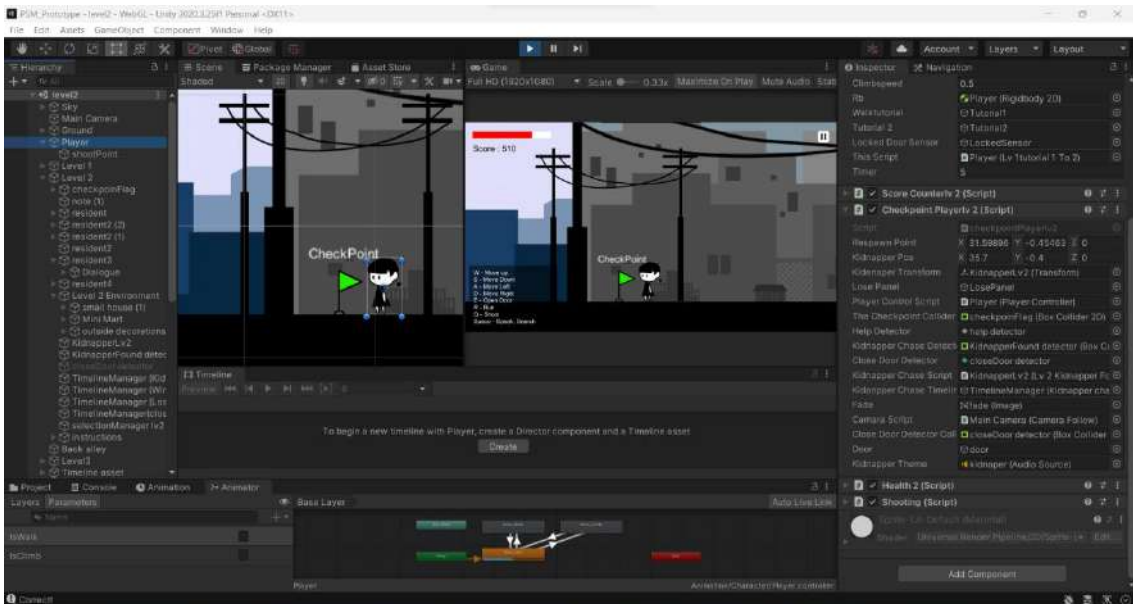


Figure 4.86 Respawn Point in Level 2

Figure 4.87 shows the checkpointPlayerlv2 Script in Level 2. The script is used to set the respawn point for players when players collide it. Players are able to respawn by click the Replay button on the lose scene.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.Playables;
6
7  public class checkpointPlayerlv2 : MonoBehaviour
8  {
9      public Vector3 respawnPoint;
10     public Vector3 kidnapperPos;
11     public Transform kidnapperTransform;
12     public GameObject losePanel;
13     public PlayerController playerController;
14
15     public BoxCollider2D theCheckpointCollider;
16     public GameObject helpDetector;
17     public BoxCollider2D kidnapperChaseDetectorCollider;
18     public GameObject closeDoorDetector;
19     public I2KIDnapperFollowPlayer kidnapperChaseScript;
20     public GameObject kidnapperChaseTimeline;
21     public Image fade;
22     public CameraFollow cameraScript;
23     public BoxCollider2D closeDoorDetectorCollider;
24     public GameObject door;
25
26     public AudioSource kidnapperTheme;
27
28     // Start is called before the first frame update
29     void Start()
30     {
31         respawnPoint = transform.position;
32         kidnapperPos = kidnapperTransform.position;
33     }
34
35     // Update is called once per frame
36     void Update()
37
38     {
39     }
40
41     private void OnTriggerEnter2D(Collider2D collision)
42     {
43         if(collision.gameObject.CompareTag("checkpointMini"))
44         {
45             respawnPoint = transform.position;
46             theCheckpointCollider.enabled = false;
47         }
48     }
49
50     public void Respawn()
51     {
52         transform.position = respawnPoint;
53         kidnapperTransform.position = kidnapperPos;
54         fade.transform.localScale = new Vector3(0, 0, 0);
55         playerController.enabled = true;
56
57         helpDetector.SetActive(false);
58         kidnapperChaseDetectorCollider.enabled = true;
59         closeDoorDetector.SetActive(false);
60         kidnapperChaseScript.enabled = false;
61         kidnapperChaseTimeline.SetActive(true);
62         losePanel.SetActive(false);
63         cameraScript.enabled = true;
64         closeDoorDetectorCollider.enabled = true;
65         door.SetActive(false);
66         kidnapperTheme.enabled = false;
67     }
68
69

```

Figure 4.87 checkpointPlayerlv2 Script in Level 2

Figure 4.88 shows the Kidnapper Harassment Timeline in Level 2. The timeline play the animation that the kidnapper harassed Anne by asking Anne play with him.

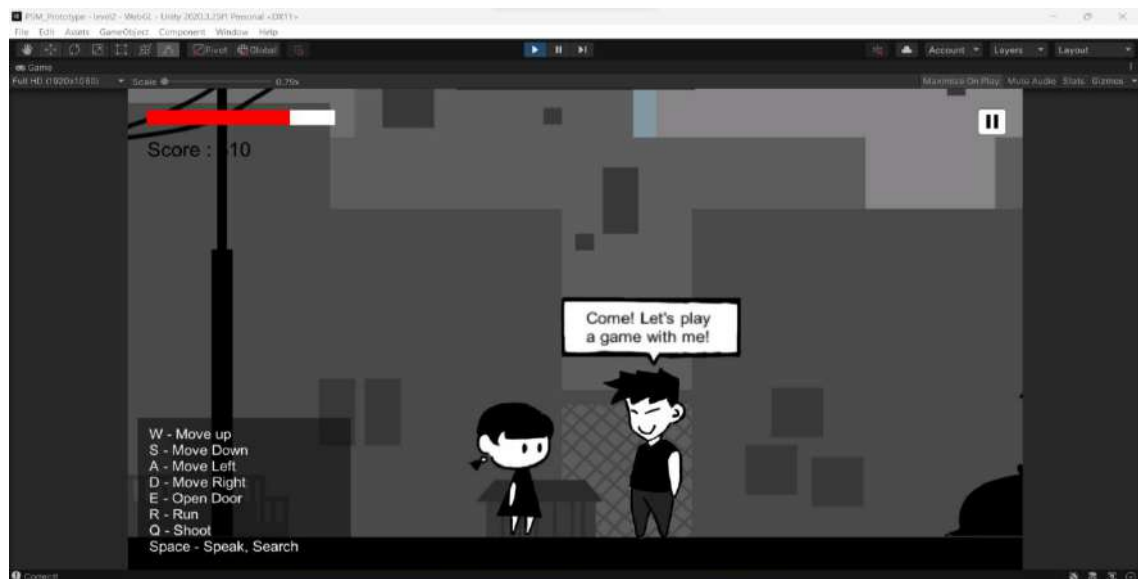


Figure 4.88 Kidnapper Harassment Timeline in Level 2

Figure 4.89 shows the Kidnapper Harassment Timeline Collider in Level 2. When player collide the collider, the Kidnapper Harassment Timeline will be played.

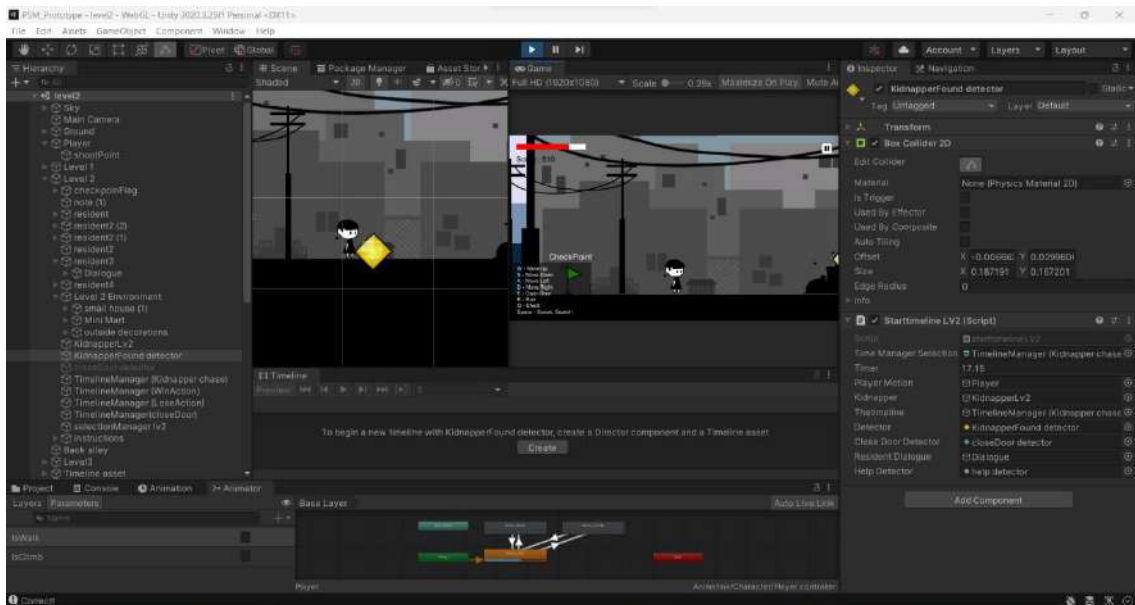


Figure 4.89 Kidnapper Harassment Timeline Collider in Level 2

Figure 4.90 shows the kidnapper chase in Level 2. After the Kidnapper Harassment Timeline finish, players will be chased by the kidnapper. Players are required to run as fast as possible and get help.



Figure 4.90 Kidnapper Chase in Level 2

Figure 4.91 shows the lv2KidnapperFollowPlayer Script in Level 2. If the distance between players and kidnapper is more than 0.3f, the kidnapper will keep approaching players. If the distance between players and kidnapper is less than 0.3f, then lose panel will be shown.

```

lv2KidnapperFollowPlayer.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class lv2KidnapperFollowPlayer : MonoBehaviour
{
    public Transform target;
    public float speed;
    public GameObject losePanel;
    public float argoRange;
    public float distoPlayer;
    private bool iscatch = false;

    public PlayerController playerControlScript;

    Rigidbody2D rb;
    BoxCollider2D col;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        col = GetComponent<BoxCollider2D>();
    }

    void Update()
    {
        distoPlayer = Vector2.Distance(transform.position, target.position);
        if (distoPlayer > argoRange)
        {
            Chase();
        }
    }

    void Chase()
    {
        if (transform.position.x > target.position.x)
        {
            rb.velocity = new Vector2(-speed, 0);
        }
    }

    void lose()
    {
        if (iscatch)
        {
            Time.timeScale = 0;
            playerControlScript.enabled = false;
            losePanel.SetActive(true);
            iscatch = false;
        }
    }
}

```

Figure 4.91 lv2KidnapperFollowPlayer Script in Level 2

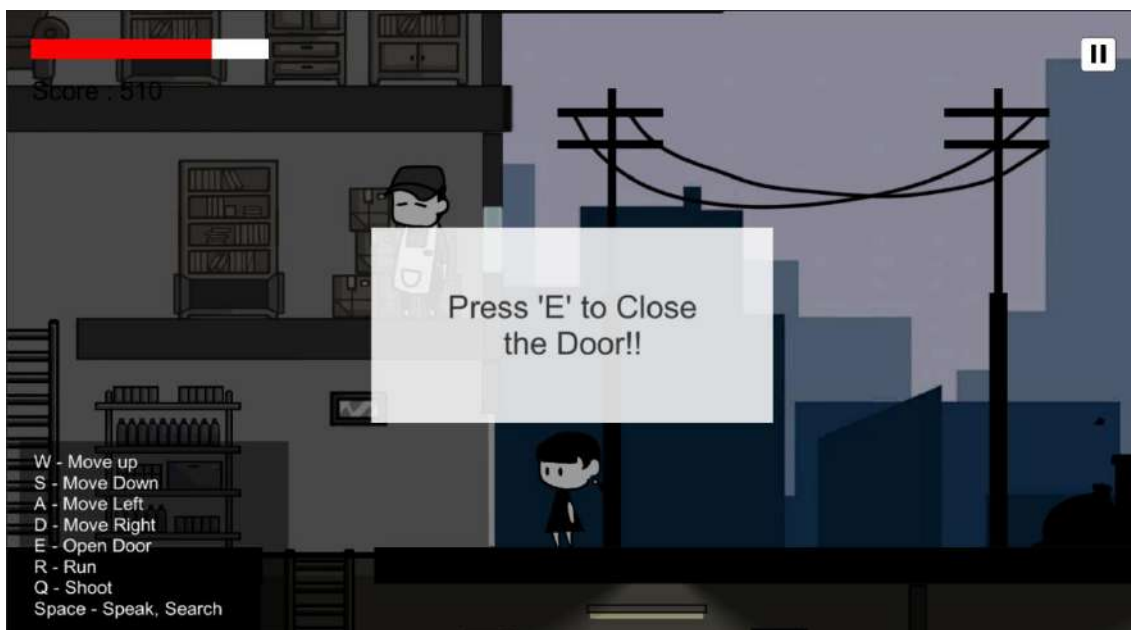


Figure 4.92 Close Door Instruction in Level 2

Figure 4.92 shows the Close Door Instruction in Level 2. On the way chasing by kidnapper, if players collide with the collider, the instruction will be shown and players are able to close the door back to stop the kidnapper approach.

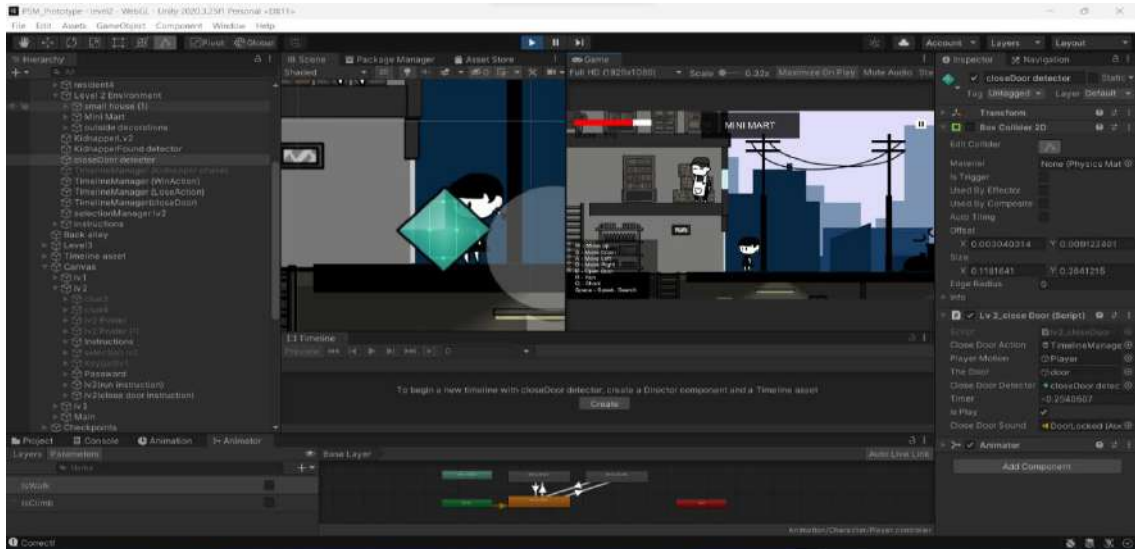


Figure 4.93 Close Door Instruction Collider in Level 2

Figure 4.94 shows the Selection Panel in Level 2. There is a question ask players whether ask help or do not and 2 buttons are provided which are Yes and No. If players choose Yes, then the win timeline will be played, else if players choose No, then the lose timeline will be played.

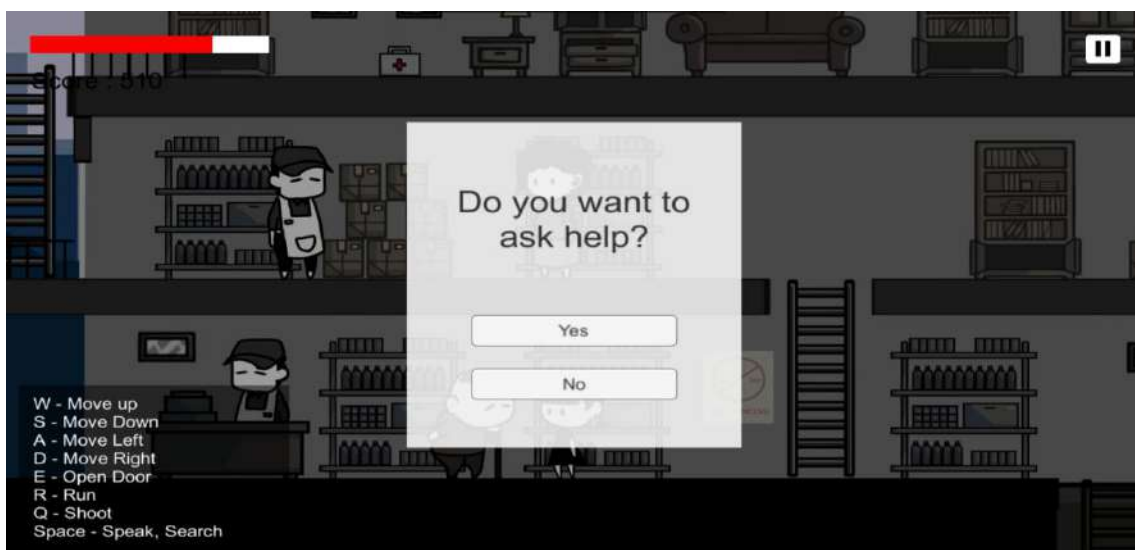


Figure 4.94 Selection Panel in Level 2

Figure 4.95 shows the starttimelineLV2 script Script in Level 2. When players collide the collider, the script will run and start played the Kidnapper Harassment Timeline and enable the kidnapper follow players script to chase the players.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.Playables;
6
7 public class startTimelineLV2 : MonoBehaviour
8 {
9     public PlayableDirector TimeManagerSelectionLV1;
10    public float timer;
11    public GameObject playerMotion;
12    public GameObject kidnapper;
13    public GameObject theTimeline;
14    public GameObject detector;
15    public GameObject closeButtonDetector;
16
17    public GameObject residentDialogue;
18    public GameObject helpDetector;
19
20    private bool IsPlaying = false;
21    private bool insIsAppear = false;
22
23    void OnCollisionEnter2D(Collision2D col)
24    {
25        if (col.gameObject.CompareTag("Player"))
26        {
27            playerMotion.GetComponent<PlayerController>().enabled = false;
28            residentDialogue.SetActive(false);
29            helpDetector.SetActive(true);
30            TimeManagerSelectionLV1.Play();
31            IsPlaying = true;
32        }
33    }
34
35
36    public void Update()
37    {
38        if (IsPlaying)
39        {
40            timer -= Time.deltaTime;
41            if (timer <= 0)
42            {
43                Time.timeScale = 0;
44                Destroy(theTimeline);
45                detector.GetComponent<BoxCollider2D>().enabled = false;
46                insIsAppear = true;
47                IsPlaying = false;
48            }
49        }
50
51        if (insIsAppear)
52        {
53            Time.timeScale = 1;
54            playerMotion.GetComponent<PlayerController>().enabled = true;
55            kidnapper.GetComponent<LV2KidnapperFollowPlayer>().enabled = true;
56            closeButtonDetector.SetActive(true);
57            insIsAppear = false;
58        }
59    }
60
}

```

Figure 4.95 startTimelineLV2 script Script in Level 2

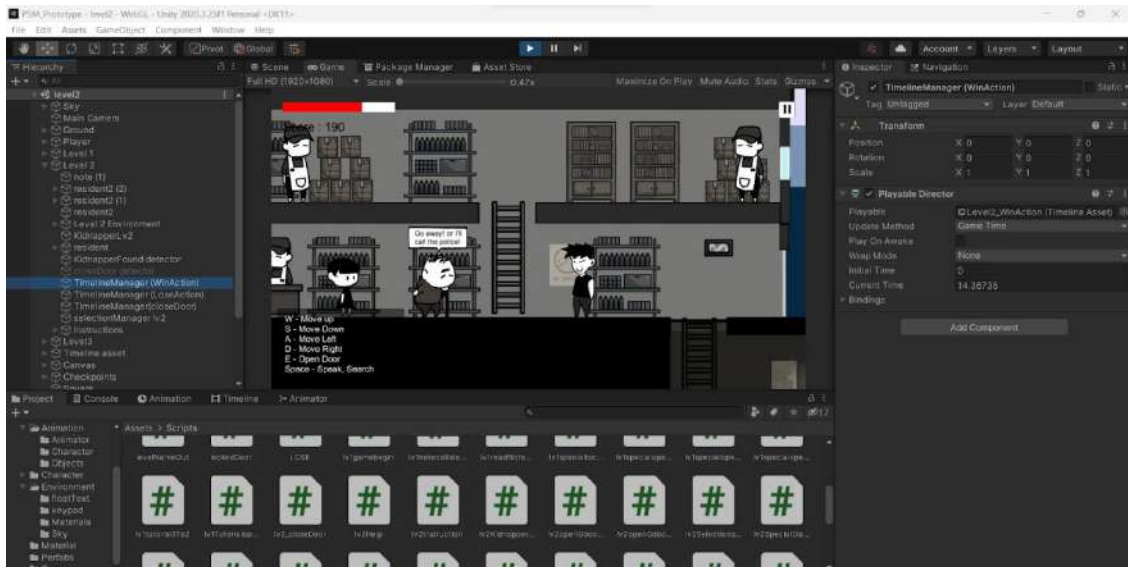


Figure 4.96 Win Timeline Level 2

Figure 4.97 shows the Checkpoint in Level 2. When players collide the checkpoint collider, then player will move to Level 3.

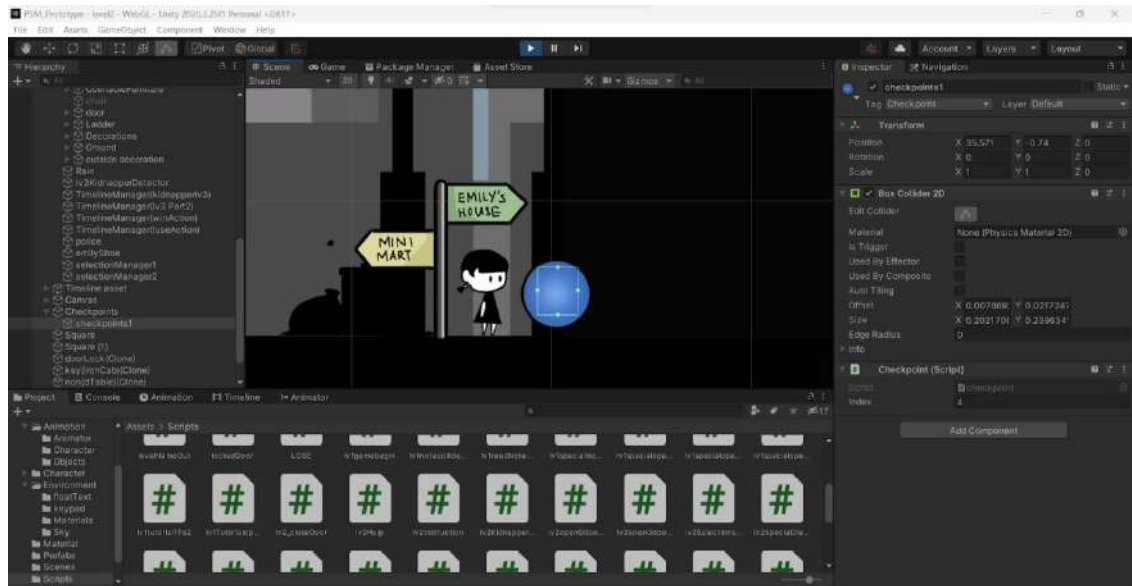


Figure 4.97 Checkpoint in Level 2

Level 3 - Emily's House

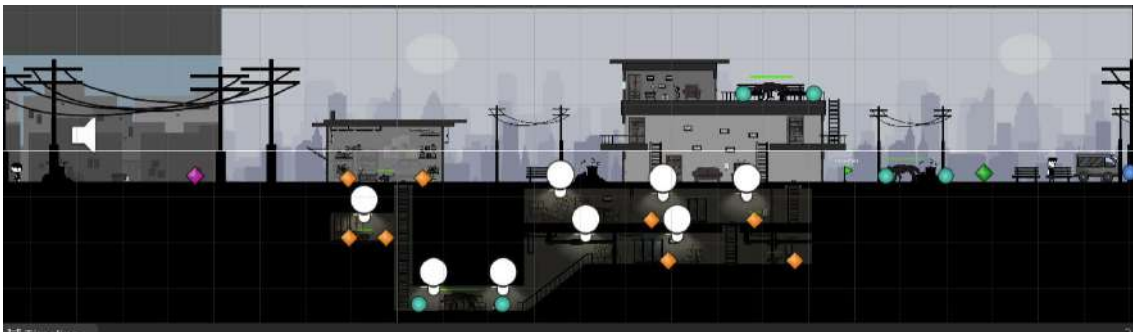


Figure 4.98 Game Environment Design of Level 3

Figure 4.99 and Figure 4.1.1 show the instruction about Emily's Shoe in Level. From the instruction, players are able to predict that Emily has been kidnapped.

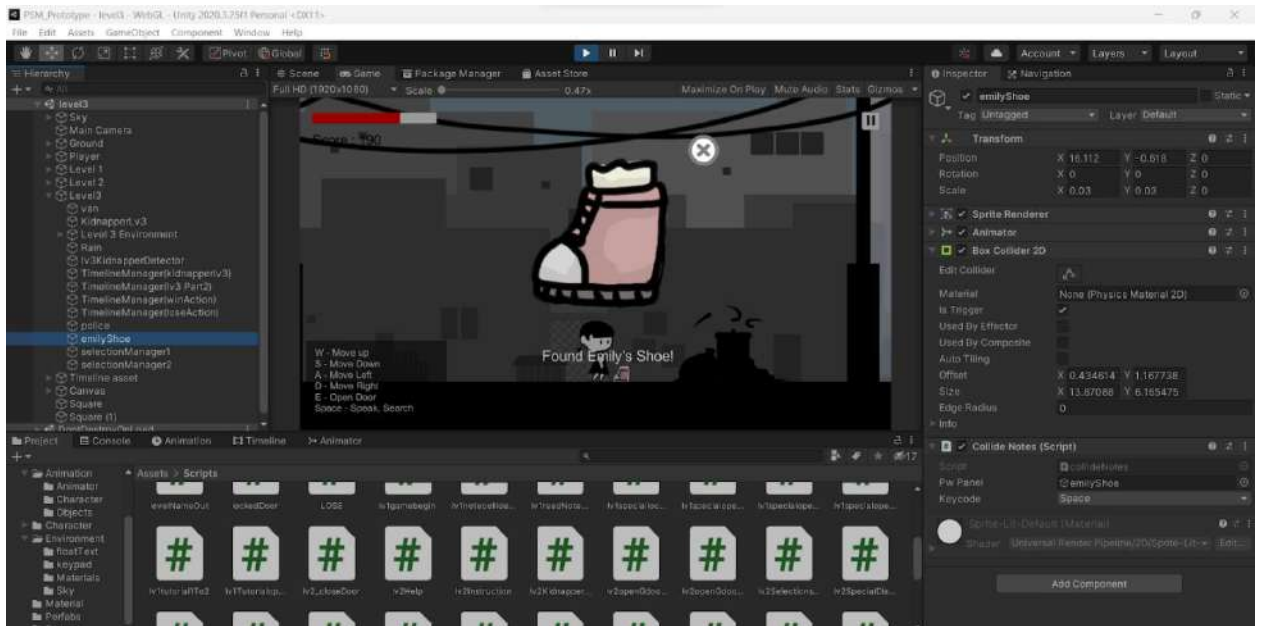


Figure 4.99 Emily Shoe Panel in Level 3

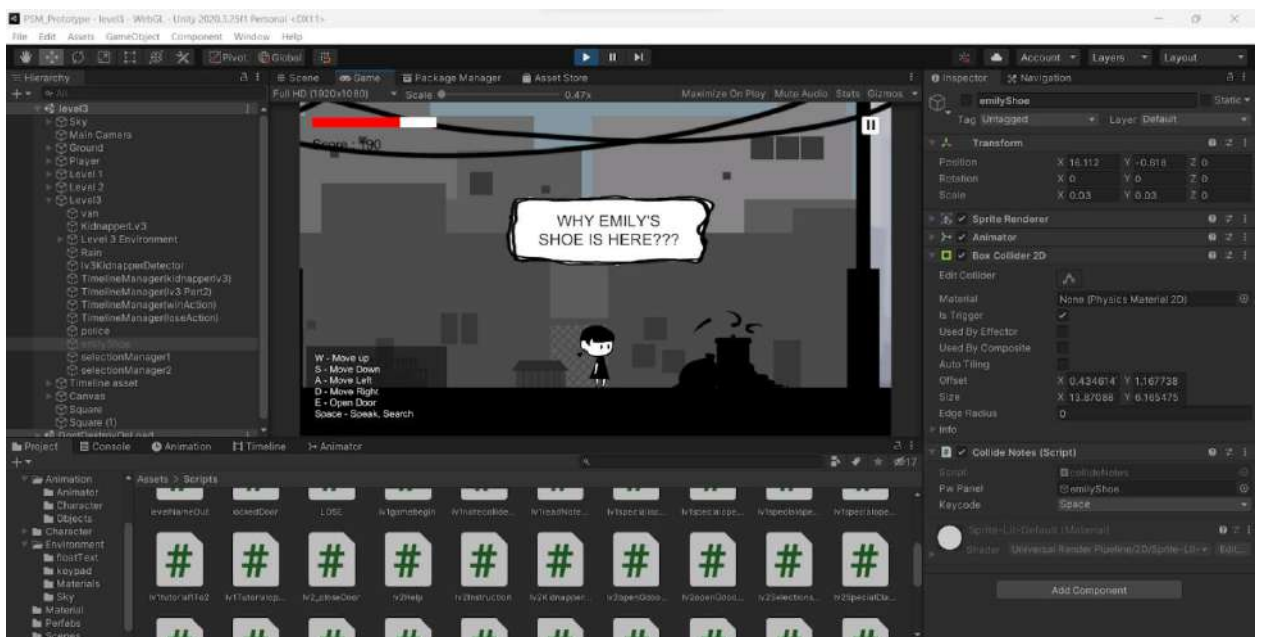


Figure 4.1.1 Instruction of Emily's Shoe Level 3

Figure 4.1.2 shows the Picture About “Do Not Ride Stranger’s Car” in Level 3. When players collider the collider and press SPACE, the message panel will be shown. The message is teach players do not ride stranger’s car.

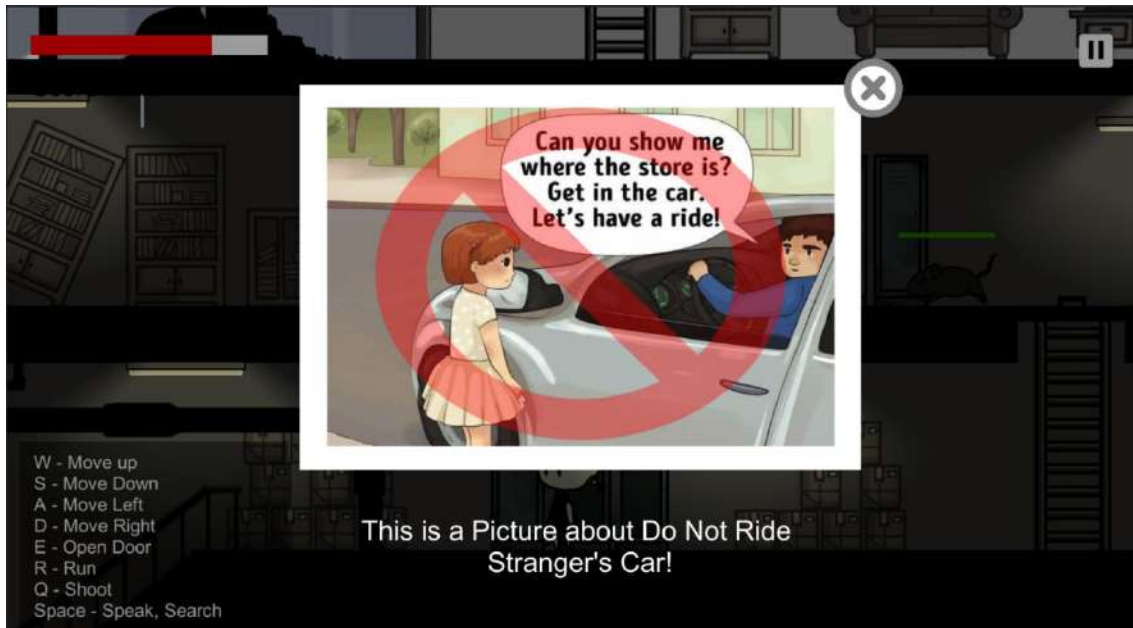


Figure 4.1.2 Picture About “Do Not Ride Stranger’s Car” in Level 3

Figure 4.1.3 shows the News About “Kidnapping fact in Melaka” in Level 3. When players collider the collider and press SPACE, the message panel will be shown. The news is important to increase the awareness of kidnapping in players.



Figure 4.1.3 News About “Kidnapping Fact in Melaka” in Level 3

Figure 4.1.4 shows the Picture About “Do Not Anything From a Stranger” in Level 3. When players collider the collider and press SPACE, the message panel will be shown. The message is teach players do not accept treats given by a stranger.

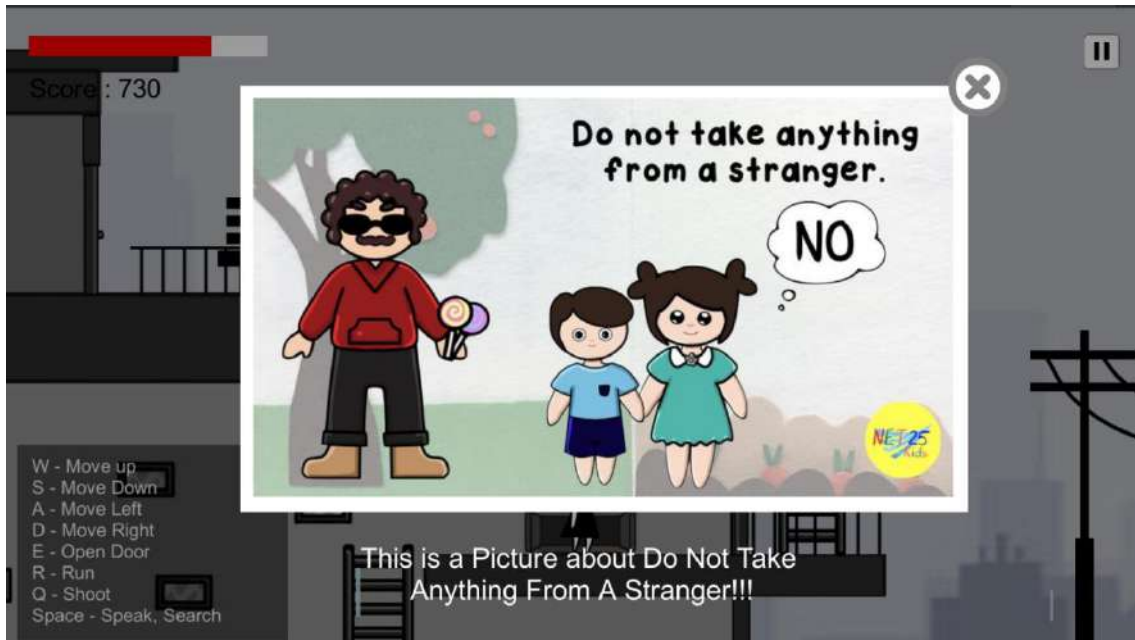


Figure 4.1.4 Picture About “Do Not Anything From a Stranger” in Level 3

Figure 4.1.5 shows the Kidnapper Timeline Detector in Level 3. When players collide the collider, the timeline will start played.

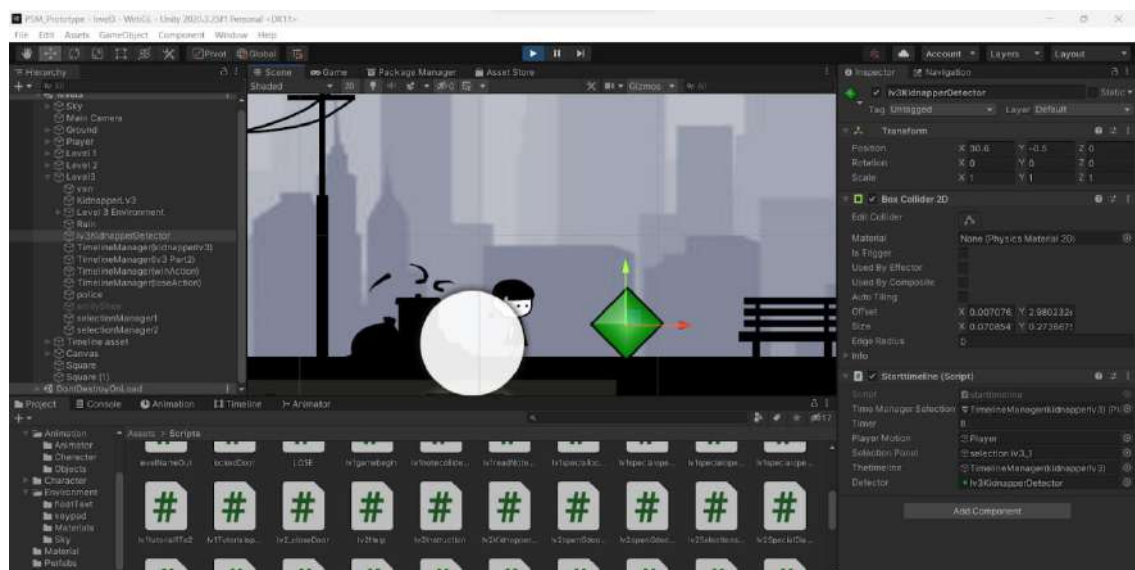


Figure 4.1.5 Kidnapper Timeline Detector in Level 3

Figure 4.1.6 shows the Kidnapper Timeline Part 1 in Level 3. In the timeline, the kidnapper trying to ask players where are they going.



Figure 4.1.6 Kidnapper Timeline Part 1 in Level 3

Figure 4.1.7 shows the Selection Panel 1 in Level 3. In the Selection 1 Panel, players are able to choose any where the purpose is to not let the kidnapper know where are players want to go.

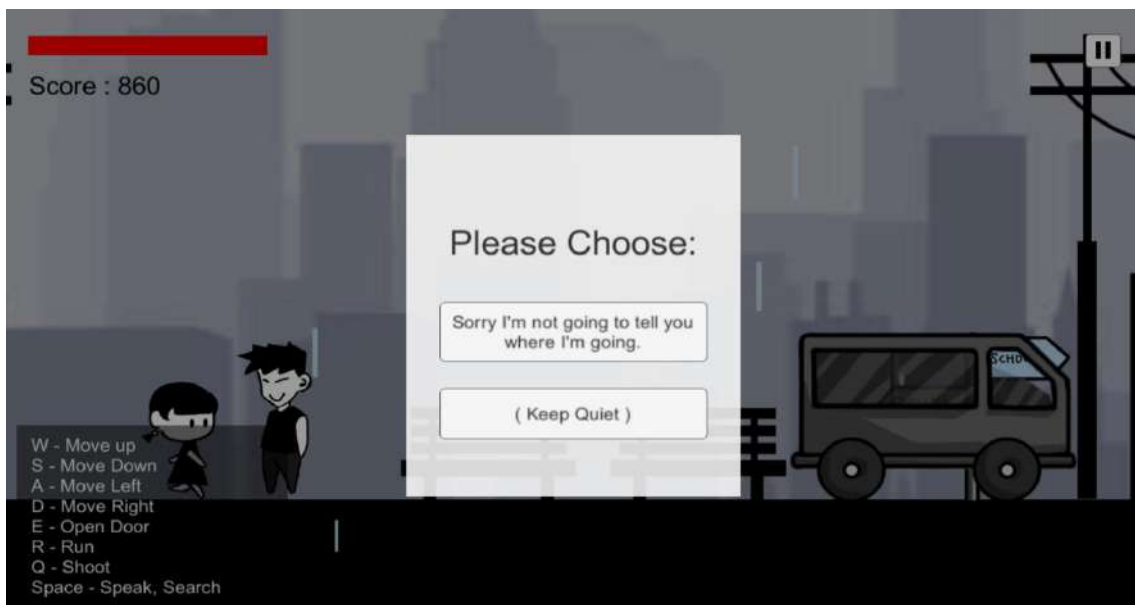


Figure 4.1.7 Selection Panel 1 in Level 3

Figure 4.1.8 shows the Kidnapper Timeline Part 2 in Level 3. The timeline play the animation where the kidnapper offer a ride to players.



Figure 4.1.8 Kidnapper Timeline Part 2 in Level 3

Figure 4.1.9 shows the Selection Panel 2 in Level 3. The selection panel ask whether players want to accept the ride or not. If players choose Yes, the the lose timeline will be player, else if player choose No, then the win timeline will be played.



Figure 4.1.9 Selection Panel 2 in Level 3

Figure 4.1.10 shows the Checkpoint in Level 3. When players collide the collider, players will be able to move to Level 4.

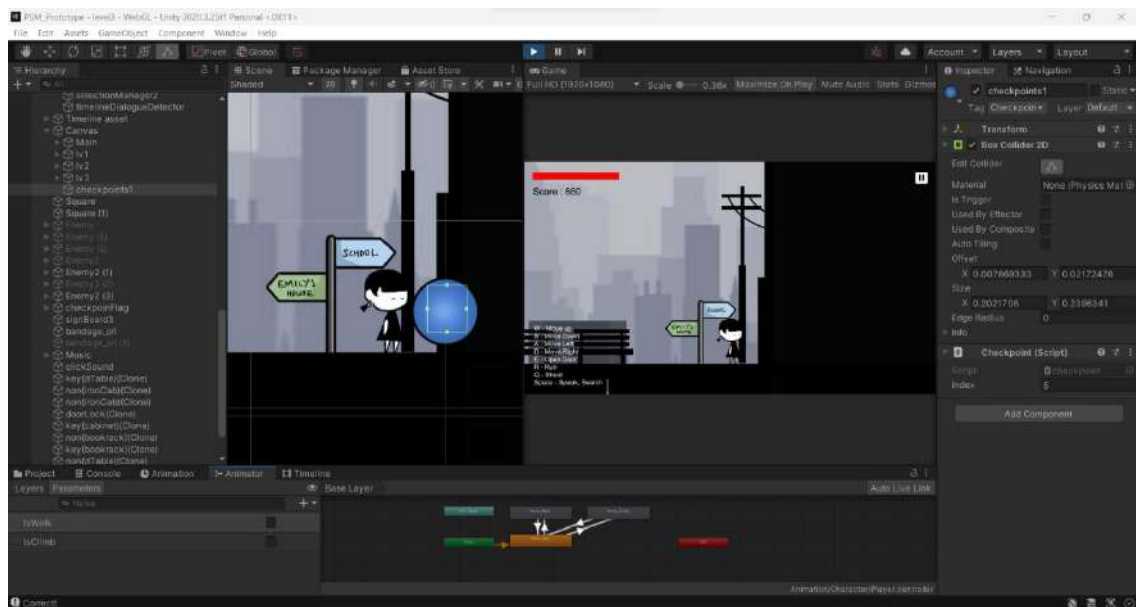


Figure 4.1.10 Checkpoint in Level 3

Level 4 - School



Figure 4.1.11 Game Environment Design of Level 4

Figure 4.1.12 shows the Find Emily Timeline in Level 4. the timeline played the animation of Anne try to find Emily in the Playground. But Anne cannot found Emily.



Figure 4.1.12 Find Emily Timeline in Level 4

Figure 4.1.13 shows the Meet Police Timeline in Level 4. The timeline is play the animation of the police meet with players and ask players to tell her if know anything about the kidnapper.



Figure 4.1.13 Meet Police Timeline in Level 4

Figure 4.1.14 and Figure 4.1.15 teach players to do something to escape from kidnappers and quickly tell the police.

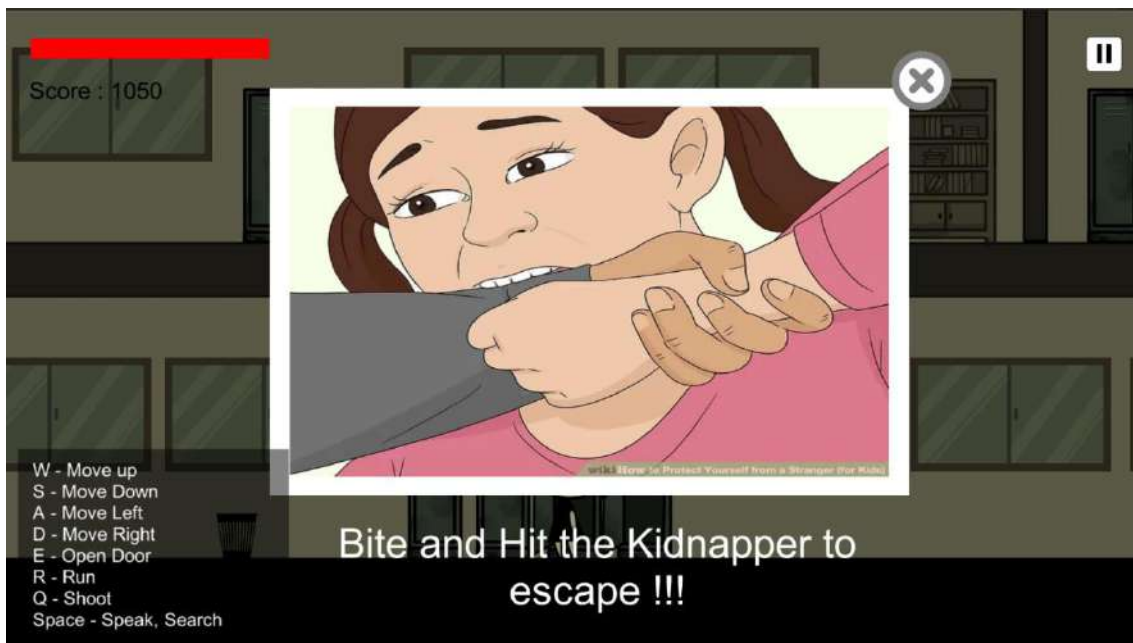


Figure 4.1.14 Picture About Do Anything To Escape in Level 4



Figure 4.1.15 Picture About Call Police If Meet the Kidnapper in Level 4

Figure 4.1.16 shows the Kidnappers Talk Timeline in Level 4. The timeline shows the animation of the conversation between kidnappers. From the timeline, players able to know the place where they hide the victims.



Figure 4.1.16 Kidnappers Talk Timeline in Level 4

Figure 4.1.17 shows the Kidnappers Chase in Level 4. After finish the Kidnapper talk Timeline, the kidnappers saw players and start to chase.



Figure 4.1.17 Kidnappers Chase in Level 4

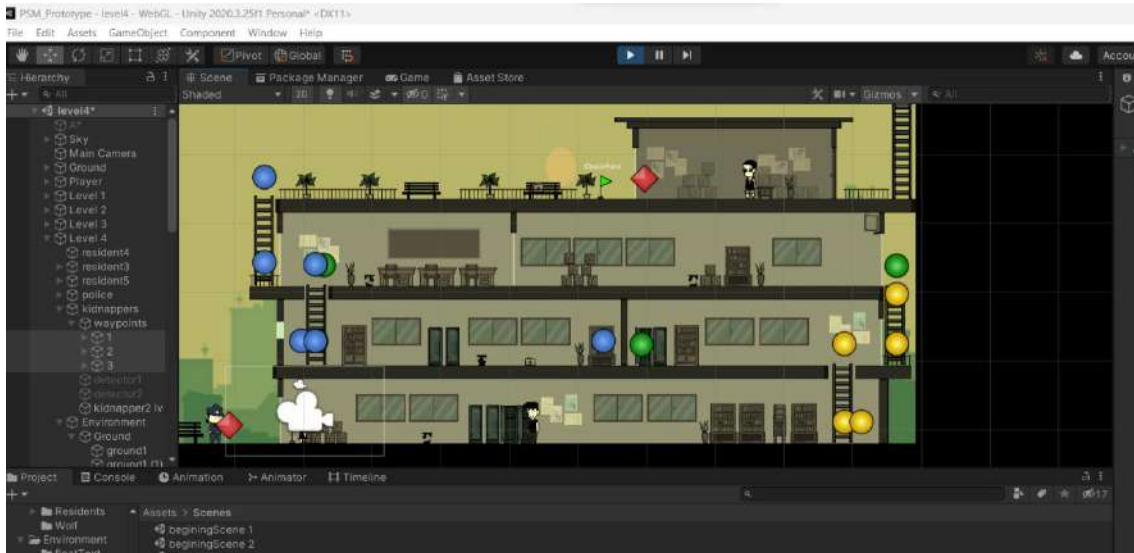


Figure 4.1.18 Kidnapper Chasing Waypoints in Level 4

Figure 4.1.19 shows the `lv4KidnapperChase` Script in Level 4. The script is used to make kidnappers chase players. When the distance between players and kidnapper is more than `float argoRange`, then kidnapper will keep chasing players follow its waypoints. If the distance between players and kidnappers is less than `float argoRange`, then Struggle Panel will shown

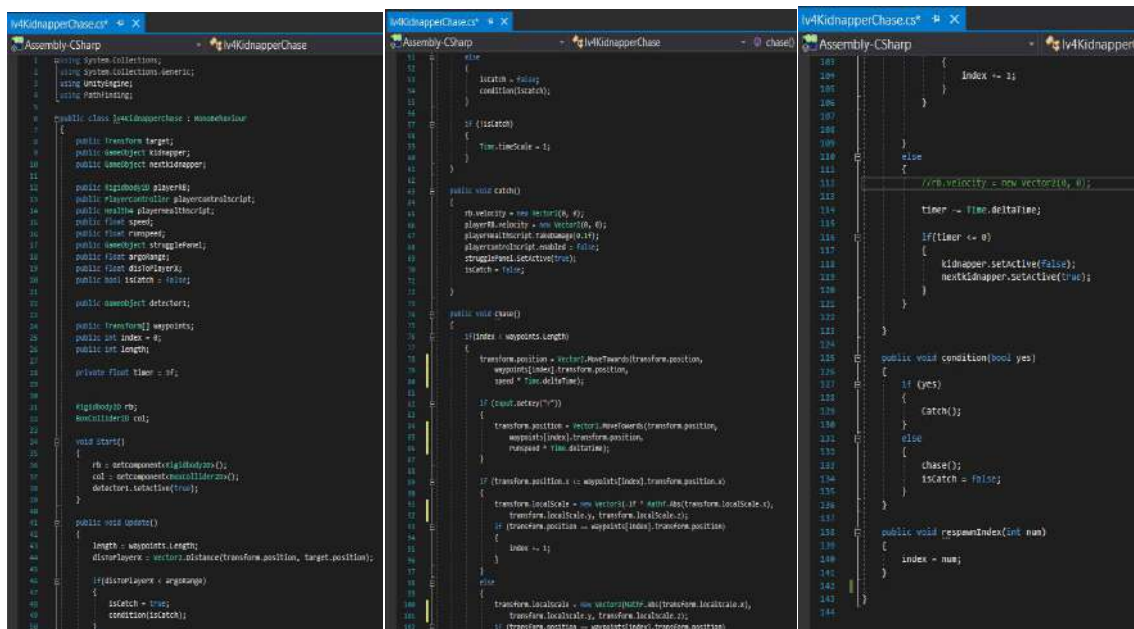


Figure 4.1.19 lv4KidnapperChase Script in Level 4

Figure 4.1.20 and Figure 4.1.21 show the Struggle Panel and lv4Struggle Script in Level 4. When distance between players and kidnapper is less than 0.3, then the Struggle Panel will be shown. Players required keep clicking the X button until the circular green bar on the Struggle Panel is full.



Figure 4.1.20 Struggle Panel in Level 4

```

lv4Struggle.cs
Assembly-CSharp
lv4Struggle
progressBar

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6
7 public class lv4Struggle : MonoBehaviour
8
9     public Image progressBar;
10    public float progressAmount = 0f;
11    public GameObject sliderPanel;
12    public PlayerController playerControlScript;
13    public lv4KidnapperChase kidnapperChaseScript;
14    public GameObject player;
15    public GameObject kidnapper;
16    public Vector3 initPos;
17
18    void Start()
19    {
20        progressBar.fillAmount = 0f;
21    }
22
23    void Update()
24    {
25        initPos = kidnapper.transform.position;
26        progressBar.fillAmount = progressAmount / 100f;
27
28        if (Input.GetKeyDown("x"))
29        {
30            progressAmount += 5;
31            progressAmount = Mathf.Clamp(progressAmount, 0, 100);
32
33            if (progressBar.fillAmount >= 1f)
34            {
35                Time.timeScale = 1;
36                playerControlScript.enabled = true;
37                progressBar.fillAmount = 0f;
38                progressAmount = 0f;
39
40                if (kidnapper.transform.position.x > player.transform.position.x)
41                {
42                    kidnapper.transform.position = new Vector3(initPos.x + 1f, initPos.y, initPos.z);
43                    player.transform.localScale = new Vector3(0.15f, 0.15f, 0);
44                }
45                else
46                {
47                    kidnapper.transform.position = new Vector3(initPos.x - 1f, initPos.y, initPos.z);
48                    player.transform.localScale = new Vector3(-0.15f, 0.15f, 0);
49                }
50                bool yes = false;
51                kidnapperChaseScript.condition(yes);
52                sliderPanel.SetActive(false);

```

Figure 4.1.21 lv4Struggle Script in Level 4

Figure 4.1.22 shows the Selection 1 Panel in Level 4. After players chased by kidnapper and meet the police again, the selection panel will be shown. The Selection 1 Panel is ask players whether want to tel the police about the kidnappers. If players choose Yes, Police Question 1 timeline will be played, else if player choose No, then it will play the lose timeline.

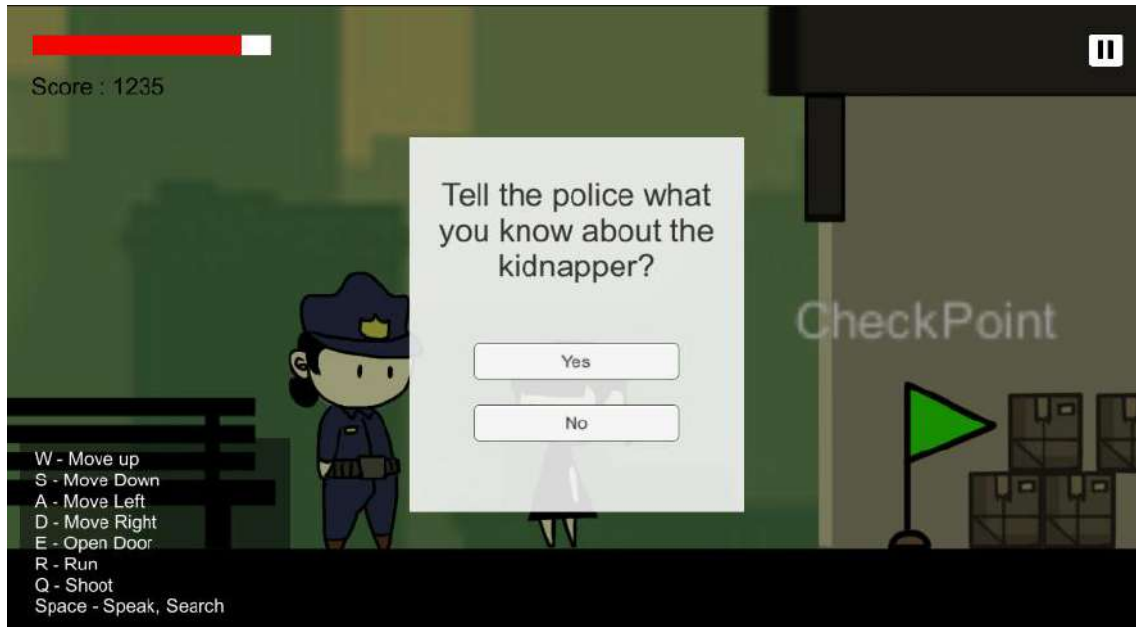


Figure 4.1.22 Selection 1 Panel in Level 4

Figure 4.1.23, Figure 4.1.24 and Figure 4.1.25 show the Selection 2 Panel in Level 4, Selection 3 Panel in Level 4, and Selection 4 Panel in Level 4. Selection 2 and 3 Panel is to choose the correct suspect of the kidnapper, while Selection 4 Panel is to choose the correct place that the kidnappers hide the victims. If there is wrong answer in one of three selections, the lose panel will activate else if players successfully answer all the selections correctly, then the win panel will be activate.

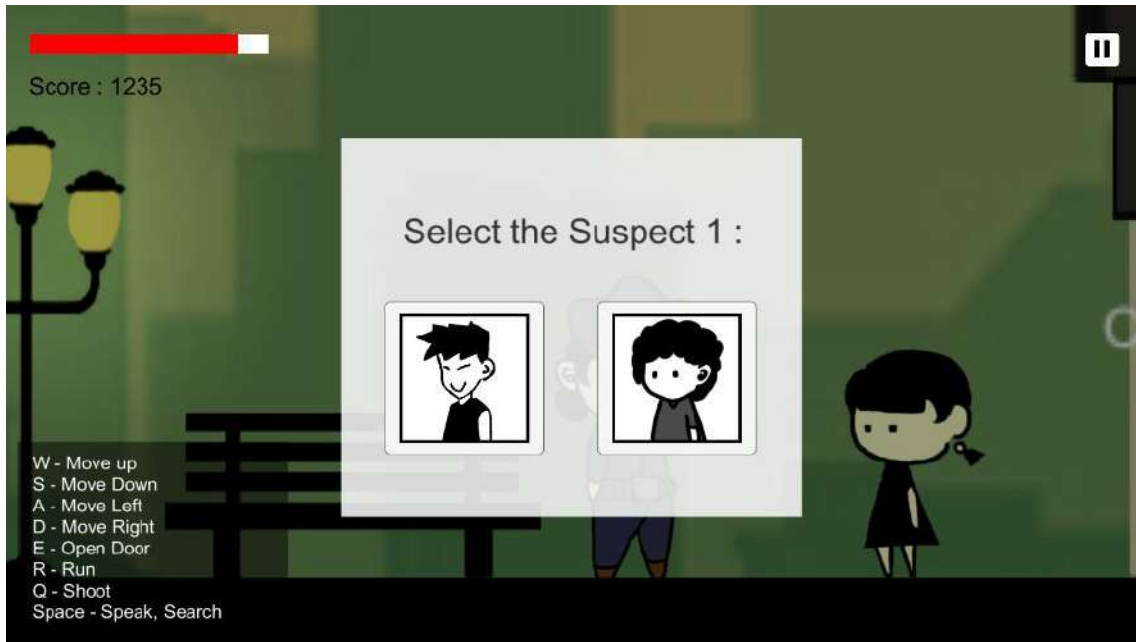


Figure 4.1.23 Selection 2 Panel in Level 4

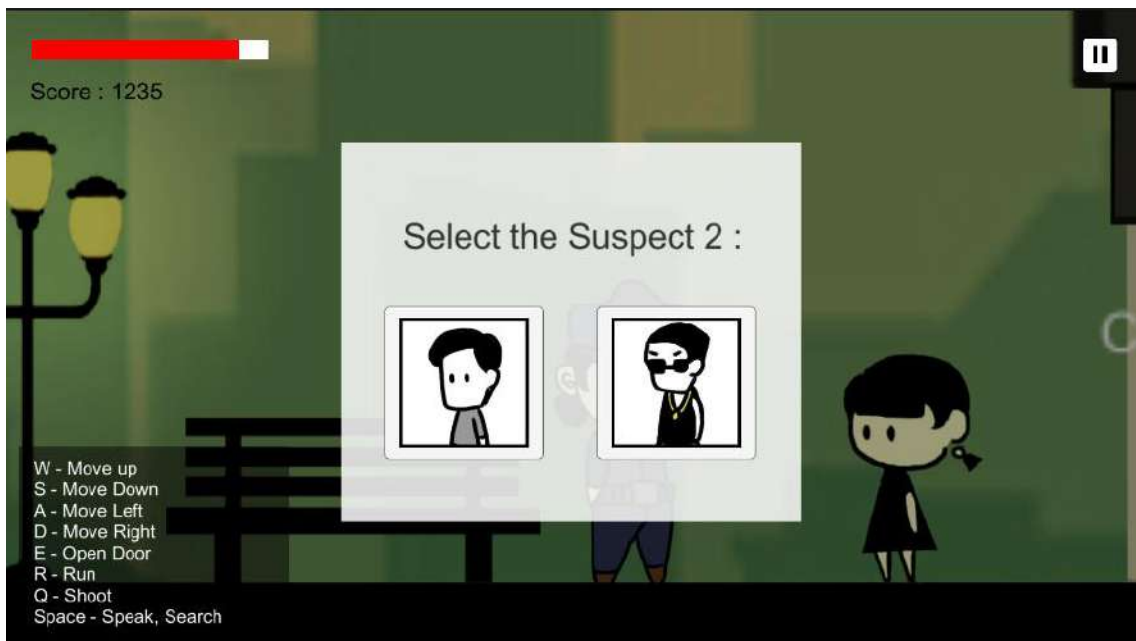


Figure 4.1.24 Selection 3 Panel in Level 4



Figure 4.1.25 Selection 4 Panel in Level 4



Figure 4.1.26 Win Panel in Level 4



Figure 4.1.27 Lose Panel in Level 4

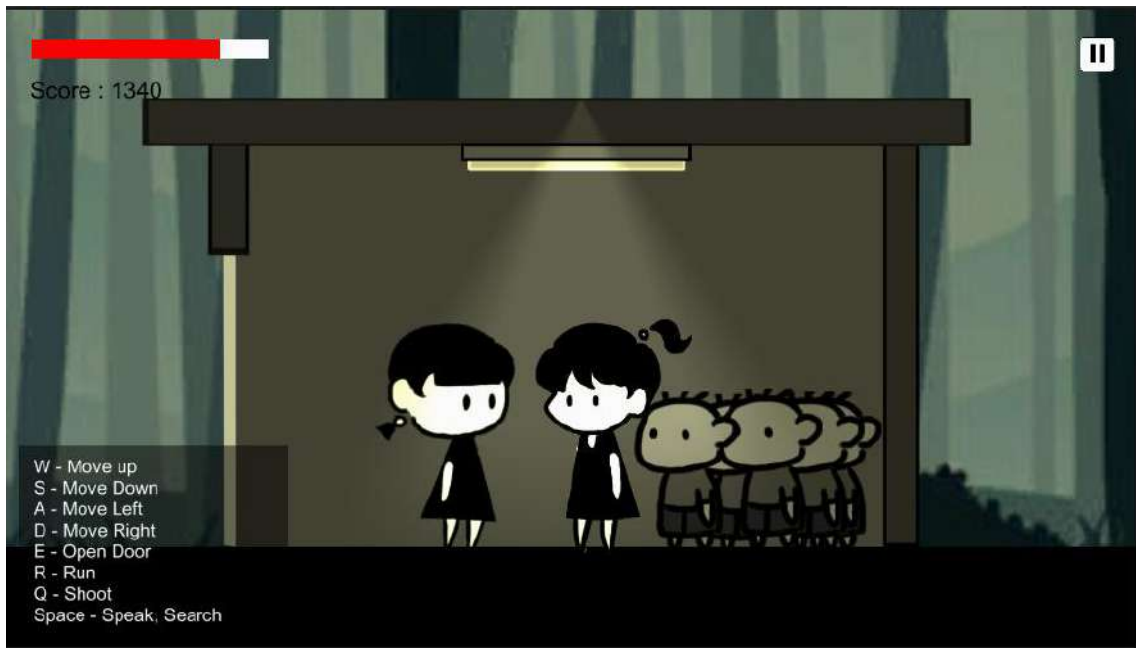


Figure 4.1.28 Game Ending Timeline in Level 4

4.2.9 Win Scene

Figure 4.1.29 shows the Win Scene of the game application. Win Scene will be shown when players complete the game. There is 1 buttons provided which is Main Menu to move to Main Menu. Players are able to view the current score they gained, new high score and the high score which is the previous high score.

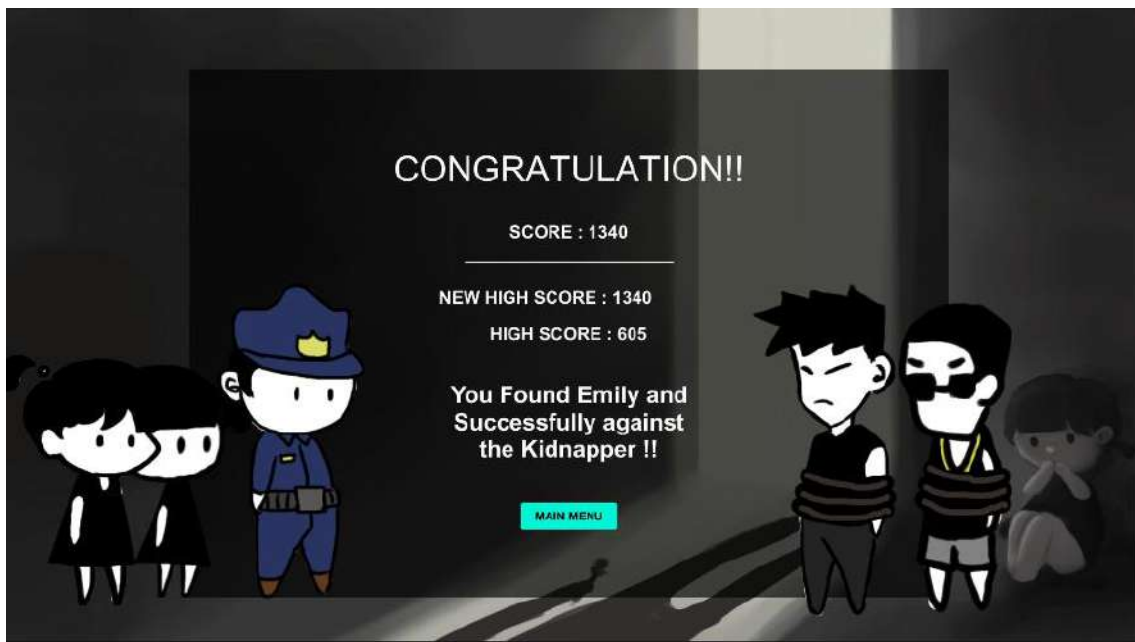


Figure 4.1.29 Win Scene Interface

Figure 4.1.30 shows the Inspector of the buttons in the Win Scene. The buttons are used on click () which with the ChangeScene script to allow the button to navigate to the correct scene after the players click the button.

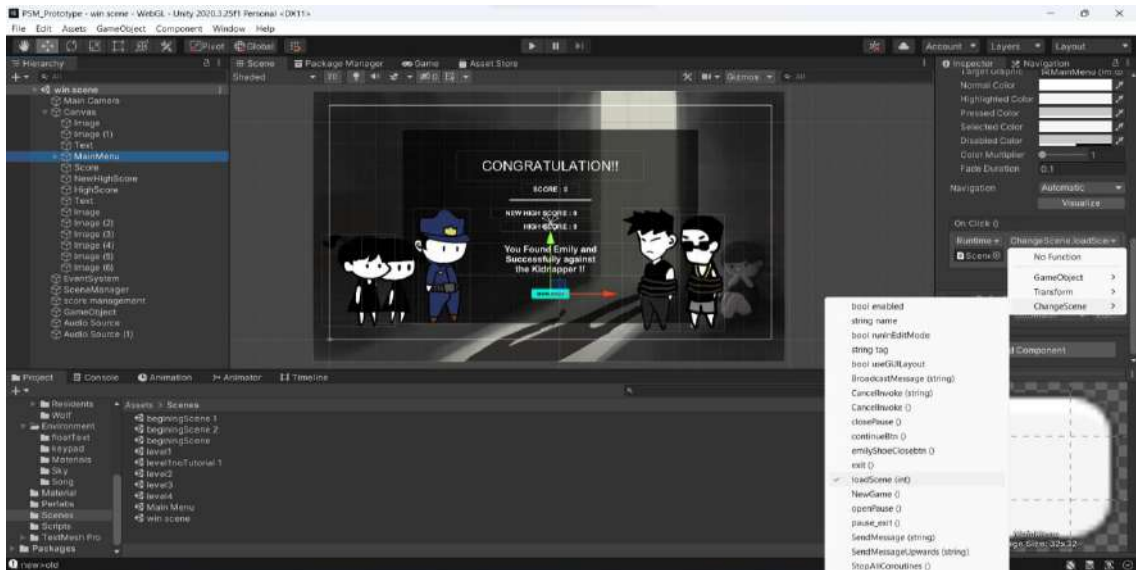


Figure 4.1.30 Inspector of the Win Scene Buttons

Figure 4.1.31 shows the highscore script used in Win Scene. The final score players gained in the game will be saved as `PlayerPrefs.SetInt("savedScore4")`. If there is players first time play the game, the score, new high score and high score will be same as saved final score. For next game, the score will be compare with previous new high score. If new final score greater than new high score, then the new high score will be placed by the new final score, else the high score no change.

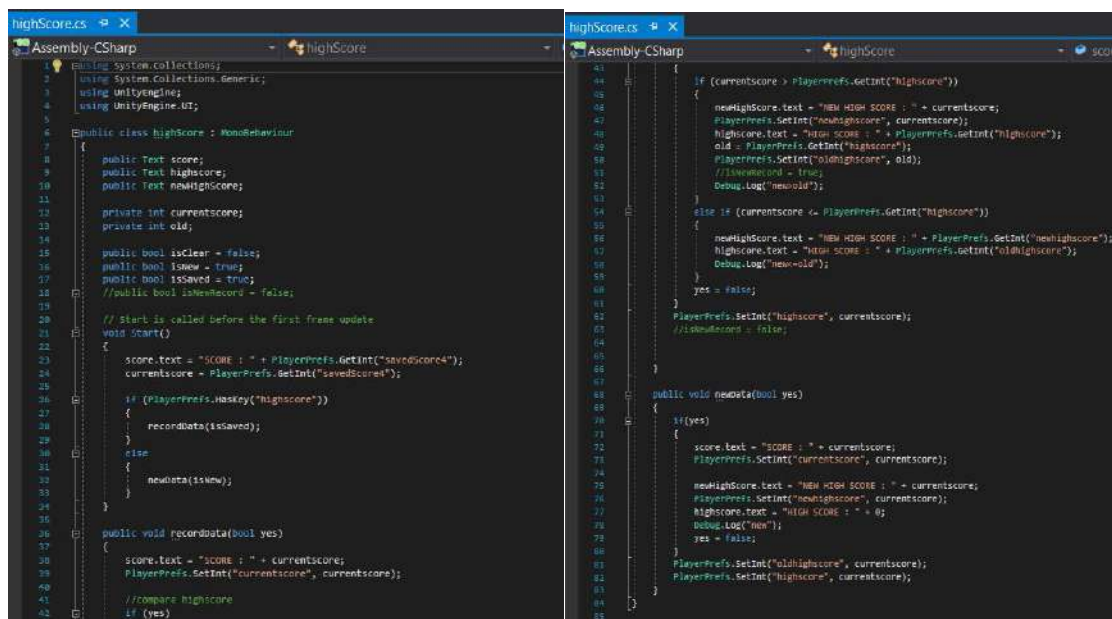


Figure 4.1.31 highscore Script in Win Scene

4.2.10 Player Controller

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7     public float walkspeed;
8     public float climbspeed;
9     private Animator anim;
10    public Rigidbody2D rb;
11
12    private float InputY;
13    private float InputX;
14
15    private bool isLadder;
16    private bool isClimb;
17
18    public void Start()
19    {
20        anim = GetComponent<Animator>();
21    }
22
23    // Update is called once per frame
24    void Update()
25    {
26        //moving left right
27        InputX = Input.GetAxis("Horizontal");
28
29        rb.velocity = new Vector2(InputX * walkspeed, rb.velocity.y);
30
31        if(InputX > 0.01f)
32        {
33            transform.localScale = new Vector2(0.15f, 0.15f);
34        }
35        else if (InputX < -0.01f)
36        {
37            transform.localScale = new Vector2(-0.15f, 0.15f);
38        }
39
40        anim.SetBool("IsWalk", InputX != 0);
41
42        InputY = Input.GetAxis("Vertical");
43
44        if(isLadder && Mathf.Abs(InputY) > 0f)
45        {
46            isClimb = true;
47            anim.SetBool("IsClimb", true);
48        }
49        else
50        {
51            anim.SetBool("IsClimb", false);
52        }
53    }
54
55    private void FixedUpdate()
56    {
57        if(isClimb)
58        {
59            rb.gravityScale = 0f;
60            rb.velocity = new Vector2(rb.velocity.x, InputY * climbspeed);
61        }
62        else
63        {
64            rb.gravityScale = 1f;
65        }
66    }
67
68    public void OnTriggerEnter2D(Collider2D collision)
69    {
70        if (collision.CompareTag("Ladder"))
71        {
72            isLadder = true;
73        }
74    }
75
76    private void OnTriggerExit2D(Collider2D collision)
77    {
78        if (collision.CompareTag("Ladder"))
79        {
80            isLadder = false;
81            isClimb = false;
82        }
83    }
84
85 }
```

Figure 4.1.32 PlayerController Script

```
Shooting.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Shooting : MonoBehaviour
6 {
7     public Transform shootPoint;
8     public GameObject bullet;
9
10    public AudioSource shootSound;
11
12    // Start is called before the first frame update
13    void Start()
14    {
15    }
16
17
18    // Update is called once per frame
19    void Update()
20    {
21        if(Input.GetKeyDown("q"))
22        {
23            shootSound.Play();
24            Instantiate(bullet, shootPoint.position, shootPoint.rotation);
25        }
26    }
27
28 }
```

Figure 4.1.33 Shooting Script

4.2.11 Camera

```
1 using UnityEngine;
2
3 public class CameraFollow : MonoBehaviour
4 {
5     public Transform target;
6     public Vector3 offset;
7
8     private void FixedUpdate()
9     {
10         transform.position = target.position + offset;
11     }
12 }
13
```

Figure 4.1.34 CameraFollow Script

4.2.12 Dialogue System

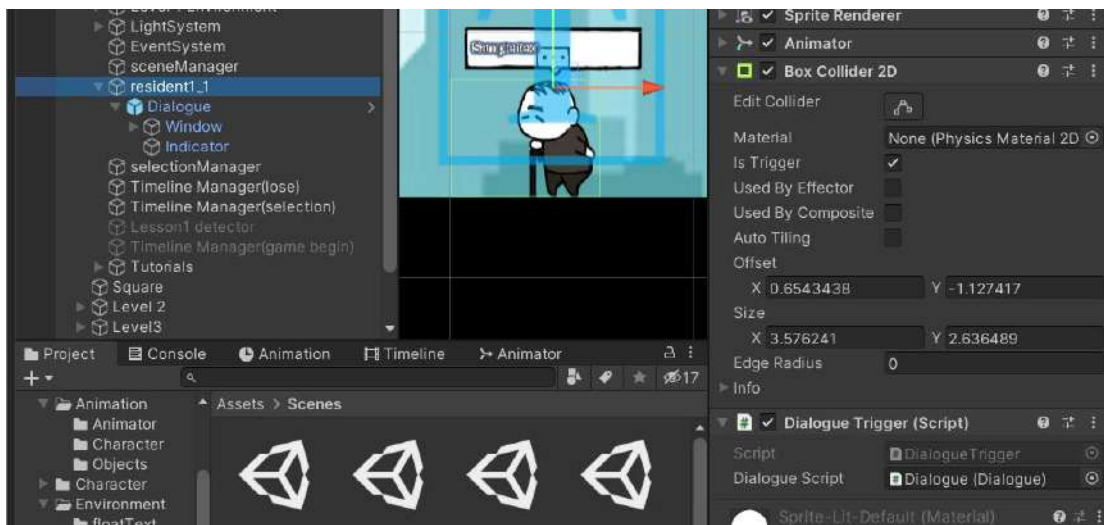


Figure 4.1.35 DialogueTrigger Script Location

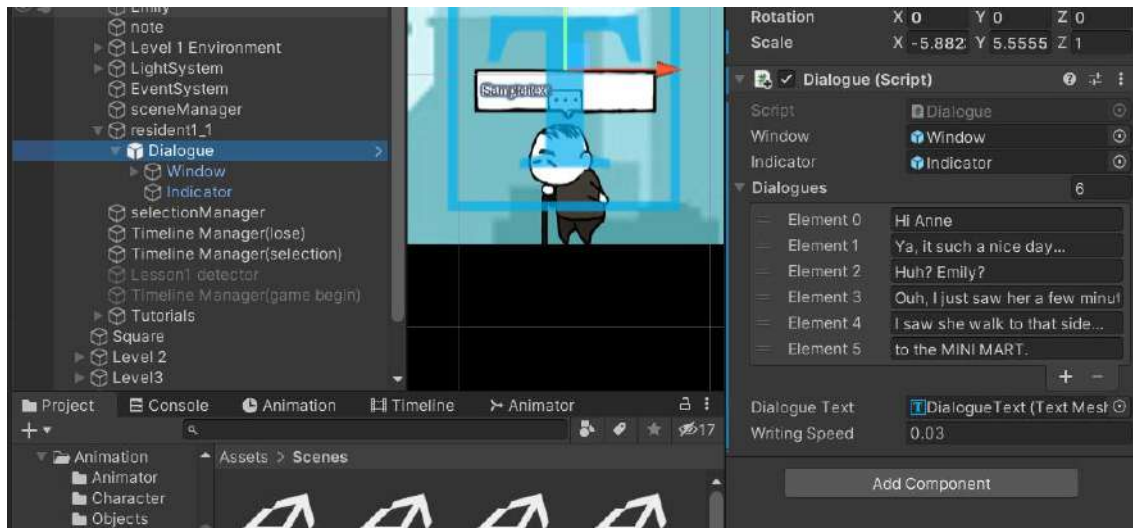


Figure 4.1.36 Dialogue Script Location

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DialogueTrigger : MonoBehaviour
6  {
7      private bool playerDetected;
8      public Dialogue dialogueScript;
9
10     private void OnTriggerEnter2D(Collider2D collision)
11     {
12         if(collision.tag == "Player")
13         {
14             playerDetected = true;
15             dialogueScript.ToggleIndicator(playerDetected);
16         }
17     }
18
19     private void OnTriggerExit2D(Collider2D collision)
20     {
21         if (collision.tag == "Player")
22         {
23             playerDetected = false;
24             dialogueScript.ToggleIndicator(playerDetected);
25             dialogueScript.EndDialogue();
26         }
27     }
28
29     private void Update()
30     {
31         if(playerDetected && Input.GetKeyDown(KeyCode.Space))
32         {
33             dialogueScript.StartDialogue();
34         }
35     }
36 }
37

```

Figure 4.1.37 DialogueTrigger Script

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using TMPPro;
5
6  public class Dialogue : MonoBehaviour
7  {
8      public GameObject window;
9      public GameObject indicator;
10     public List<string> dialogues;
11     public TMP_Text dialogueText;
12     public float writingSpeed;
13
14     private int index;
15     private int charIndex;
16
17     private bool started;
18     private bool waitForNext;
19
20     private void Awake()
21     {
22         ToggleIndicator(false);
23         ToggleWindow(false);
24     }
25
26     public void ToggleWindow(bool show)
27     {
28         window.SetActive(show);
29     }
30
31     public void ToggleIndicator(bool show)
32     {
33         indicator.SetActive(show);
34     }
35
36     public void StartDialogue()
37     {
38         if (started)
39             return;
40
41         started = true;
42         ToggleWindow(true);
43         ToggleIndicator(false);
44         GetDialogue(0);
45     }
46
47     private void GetDialogue(int i)
48     {
49         index = i;
50         charIndex = 0;
51         dialogueText.text = string.Empty;
52         StartCoroutine(Writing());
53     }
54
55     public void EndDialogue()
56     {
57         started = false;
58         waitForNext = false;
59         StopAllCoroutines();
60         ToggleWindow(false);
61     }
62
63     IEnumerator Writing()
64     {
65         yield return new WaitForSeconds(writingSpeed);
66
67         string currentDialogue = dialogues[index];
68         dialogueText.text += currentDialogue[charIndex];
69         charIndex++;
70
71         if(charIndex < currentDialogue.Length)
72         {
73             yield return new WaitForSeconds(writingSpeed);
74             StartCoroutine(Writing());
75         }
76         else
77         {
78             waitForNext = true;
79         }
80     }
81
82     private void Update()
83     {
84         if (!started)
85             return;
86         if (waitForNext)
87         {
88             waitForNext = false;
89             index++;
90             StartCoroutine(delay());
91         }
92     }
93
94     IEnumerator delay()
95     {
96         yield return new WaitForSeconds(2f);
97
98         if (index < dialogues.Count)
99         {
100             GetDialogue(index);
101         }
102         else
103         {
104             ToggleIndicator(true);
105             EndDialogue();
106         }
107     }
108
109
110
111
112

```

Figure 4.1.38 Dialogue Script

4.2.13 Open Furnitures

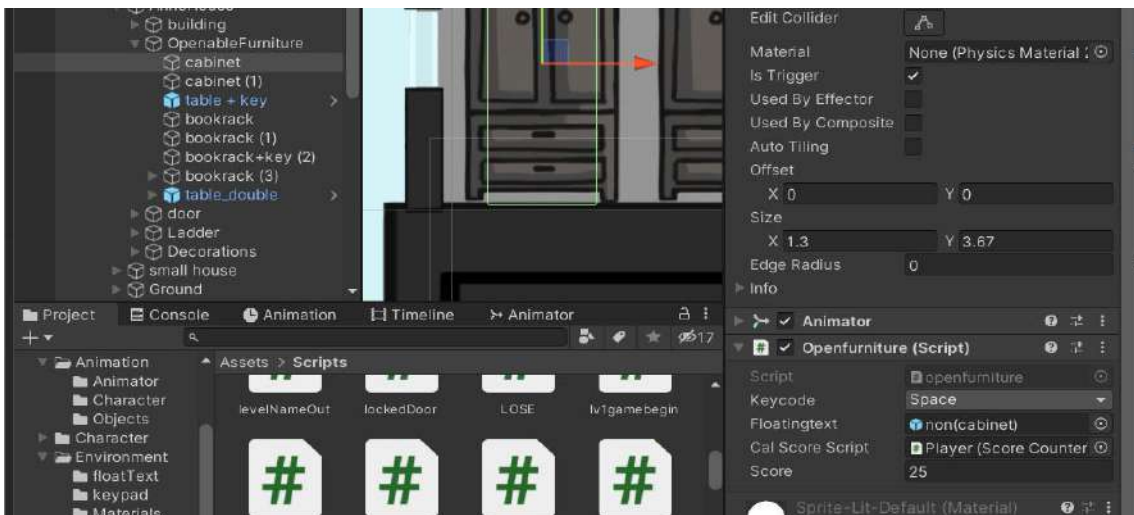


Figure 4.1.39 Openfurniture Script Location

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class openfurniture : MonoBehaviour
6  {
7      private Animator anim;
8      public KeyCode keycode;
9      private bool selected = false;
10     public GameObject floatingtext;
11     private openfurniture scriptt;
12     public ScoreCounter calScoreScript;
13     public int score;
14
15     // Start is called before the first frame update
16     void Start()
17     {
18         anim = GetComponent<Animator>();
19         scriptt = GetComponent<openfurniture>();
20     }
21
22     void Update()
23     {
24         if (selected && Input.GetKeyDown(keycode))
25         {
26             anim.SetBool("isOpen", true);
27             Instantiate(floatingtext, transform.position, Quaternion.identity);
28             calScoreScript.collectScore(score);
29             scriptt.enabled = false;
30         }
31     }
32
33     private void OnTriggerEnter2D(Collider2D collision)
34     {
35         if (collision.CompareTag("Player"))
36         {
37             anim.SetBool("isSelect", true);
38             selected = true;
39         }
40     }
41
42     private void OnTriggerExit2D(Collider2D collision)
43     {
44         if (collision.CompareTag("Player"))
45         {
46             anim.SetBool("isSelect", false);
47             selected = false;
48         }
49     }
50 }

```

Figure 4.1.40 Openfurniture Script

4.2.14 Score

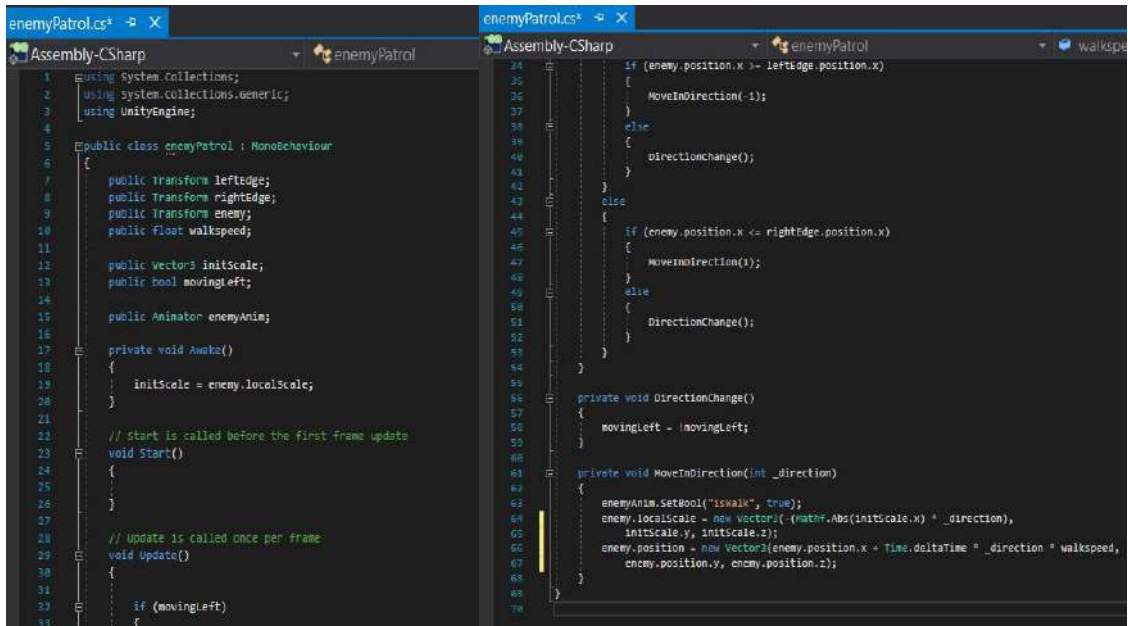
```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class ScoreCounter : MonoBehaviour
7  {
8      public Text myScore;
9      private int total = 0;
10     public int newTotal;
11
12     // Start is called before the first frame update
13     void Start()
14     {
15         myScore.text = "Score : " + total;
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21         myScore.text = "Score : " + total;
22         PlayerPrefs.SetInt("savedScore1", total);
23         newTotal = PlayerPrefs.GetInt("savedScore1");
24     }
25
26     public void collectScore(int score)
27     {
28         total += score;
29     }
30 }

```

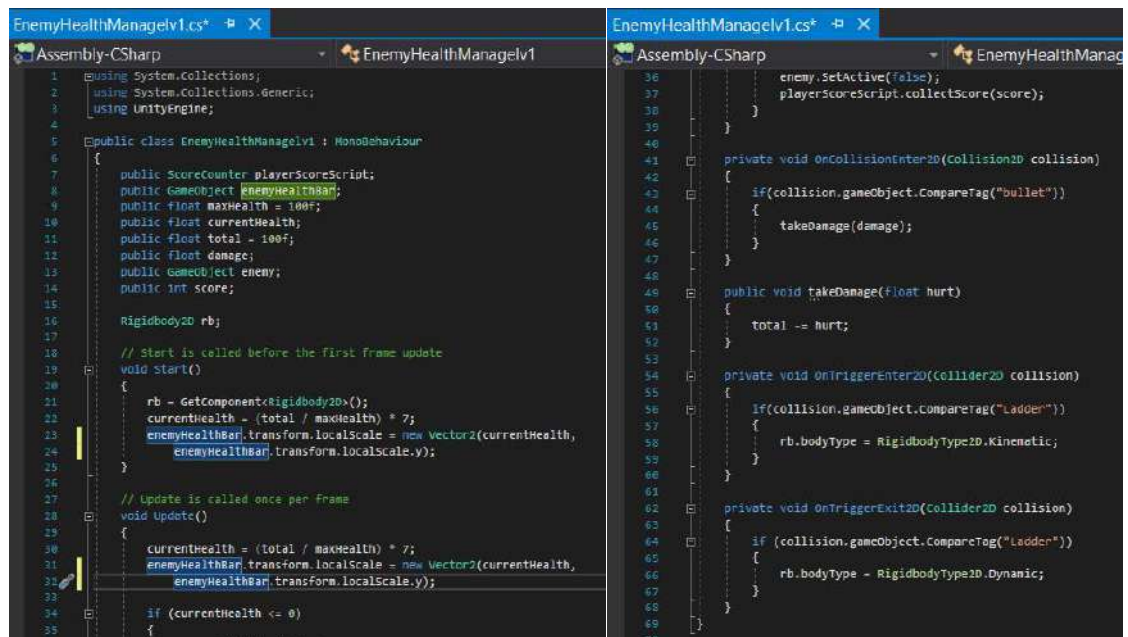
Figure 4.1.41 ScoreCounter Script

4.2.15 Enemy Behavior



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class enemyPatrol : MonoBehaviour
6 {
7     public Transform leftEdge;
8     public Transform rightEdge;
9     public Transform enemy;
10    public float walkspeed;
11
12    public Vector3 initScale;
13    public bool movingLeft;
14
15    public Animator enemyAnim;
16
17    private void Awake()
18    {
19        initScale = enemy.localScale;
20    }
21
22    // Start is called before the first frame update
23    void Start()
24    {
25
26    }
27
28    // Update is called once per frame
29    void Update()
30    {
31        if (movingLeft)
32        {
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Figure 4.1.42 enemyPatrol Script



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyHealthManager1 : MonoBehaviour
6 {
7     public ScoreCounter playerScoreScript;
8     public GameObject enemyHealthBar;
9     public float maxHealth = 100f;
10    public float currentHealth;
11    public float total = 100f;
12    public float damage;
13    public GameObject enemy;
14    public int score;
15
16    Rigidbody2D rb;
17
18    // Start is called before the first frame update
19    void Start()
20    {
21        rb = GetComponent<Rigidbody2D>();
22        currentHealth = (total / maxHealth) * 7;
23        enemyHealthBar.transform.localScale = new Vector2(currentHealth,
24        enemyHealthBar.transform.localScale.y);
25    }
26
27    // Update is called once per frame
28    void Update()
29    {
30        currentHealth = (total / maxHealth) * 7;
31        enemyHealthBar.transform.localScale = new Vector2(currentHealth,
32        enemyHealthBar.transform.localScale.y);
33    }
34    if (currentHealth <= 0)
35    {
36        enemy.SetActive(false);
37        playerScoreScript.collectScore(score);
38    }
39    }
40
41    private void OnCollisionEnter2D(Collision2D collision)
42    {
43        if (collision.gameObject.CompareTag("bullet"))
44        {
45            takeDamage(damage);
46        }
47    }
48
49    public void takeDamage(float hurt)
50    {
51        total -= hurt;
52    }
53
54    private void OnTriggerEnter2D(Collider2D collision)
55    {
56        if (collision.gameObject.CompareTag("Ladder"))
57        {
58            rb.bodyType = RigidbodyType2D.Kinematic;
59        }
60    }
61
62    private void OnTriggerExit2D(Collider2D collision)
63    {
64        if (collision.gameObject.CompareTag("Ladder"))
65        {
66            rb.bodyType = RigidbodyType2D.Dynamic;
67        }
68    }
69
70
```

Figure 4.1.43 EnemyHealthManage Script

Figure 4.1.44 shows the enemyControl Script which used to detect the players. When players enter enemy's sight, then enemy will start to chase players. When the distance

between players and enemy less than *float argoRange* then enemy start to attack players and damage players' health bar.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class enemyControlLv1 : MonoBehaviour
6 {
7     public Transform wolf;
8     public GameObject wolf;
9     public float runSpeed;
10    public float argoRange;
11    public float distoPlayer;
12    public enemyPatrol enemyPatrolScript;
13    public Health playerHealthScript;
14    public float damage;
15    public bool isChase = false;
16
17    public AudioClip rd;
18    public Animator enemyAnim;
19
20    public AudioSource walkSound;
21
22    void Start()
23    {
24    }
25
26    private void OnTriggerEnter2D(Collider2D collision)
27    {
28        if (collision.gameObject.CompareTag("Player"))
29        {
30            isChase = true;
31        }
32    }
33
34    private void OnTriggerExit2D(Collider2D collision)
35    {
36        if (collision.gameObject.CompareTag("Player"))
37        {
38            isChase = false;
39        }
40    }
41
42    void Update()
43    {
44        if (isChase)
45        {
46            enemyPatrolScript.enabled = false;
47            Chase();
48        }
49        else
50        {
51            enemyPatrolScript.enabled = true;
52            enemyAnim.SetBool("isChase", false);
53            enemyAnim.SetBool("isAttack", false);
54            enemyAnim.SetBool("isHit", true);
55
56            distoPlayer = Vector2.Distance(wolf.transform.position,
57                target.position);
58
59            if (distoPlayer <= argoRange)
60            {
61                attack();
62            }
63
64            if (distoPlayer < 2.5f)
65            {
66                walkSound.maxDistance = 60f;
67            }
68            else
69            {
70                walkSound.maxDistance = 24f;
71            }
72
73            if (!isChase)
74            {
75                Time.timeScale = 1;
76            }
77        }
78    }
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

```

Figure 4.1.44 enemyControl Script

4.2.16 Bandage

Figure 4.1.45 shows the collideBandage Script. When the bandage collider is detect the Player and SPACE is pressed, the script will pass the 50 point of health to players' health script to heal the players.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class collideBandageLv1 : MonoBehaviour
6 {
7     private Animator anim;
8     private bool selected = false;
9     public Health healthScript;
10    public KeyCode keycode;
11
12    public AudioSource healingSound;
13
14    void Start()
15    {
16        anim = GetComponent<Animator>();
17    }
18
19    void Update()
20    {
21        if (selected && Input.GetKeyDown(keycode))
22        {
23            healingSound.Play();
24            gameObject.SetActive(false);
25            healthScript.Heal(50);
26        }
27    }
28
29    private void OnTriggerEnter2D(Collider2D collision)
30    {
31        if (collision.CompareTag("Player"))
32        {
33            anim.SetBool("isSelect", true);
34            selected = true;
35        }
36    }
37
38
39    private void OnTriggerExit2D(Collider2D collision)
40    {
41        if (collision.CompareTag("Player"))
42        {
43            anim.SetBool("isSelect", false);
44            selected = false;
45        }
46    }
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 4.1.45 collideBandage Script

4.3 Evaluation Phase

In this phase, the User Acceptance Test (UAT) and Functional Acceptance Test (FAT) form is created using Google Form. The form has been tested by 15 tester. The FAT form is used to test the functional of the game application and the UAT form is used to test the player's experience of the game application.

4.3.1 Result of Functional And User Acceptance Test

Table 4.1 shows the Result of FAT and UAT form filled by 15 tester. For FAT part, the FAT is test for Main Menu, Level 1 until 4 and Win Scene. From the result of the FAT form, the functional of the game application is function well and players are able to play the game successfully. From the UAT form, the game is successfully download and played on PC. 13 tester out of 15 in total agree that the Found You 2D game is interesting and easy to understand and only 2 tester feel natural. 15 tester agree that the Found You 2D game is able to teach player about the techniques of protect themselves from being kidnapped and able to let players know about the techniques or treats that will be used by kidnapper. 14 tester out of 15 in total agree that the Found You 2D game ablr to increase the awareness of child abduction in player. From the 5 questions, all the tester are able to answer it correctly means that the testers are learnt and understood the lesson bring from the game.

Table 4.1 Result of FAT and UAT Form

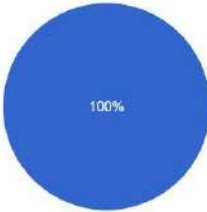
No	Question & Result
Section1 : FAT Main Menu	
1	<p>is player able to start a new game after click NEW GAME button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>Yes</p>

Table 4.1 Continued

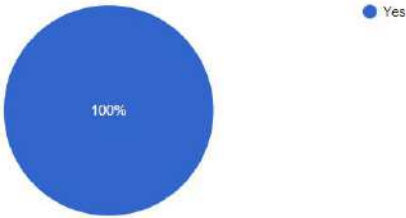
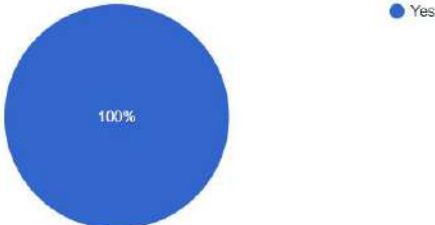
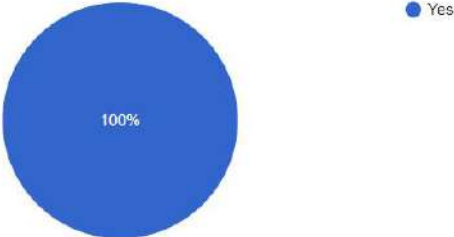

No	Question & Result
2	<p>is player able to continue the previous exit level after click CONTINUE button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p>
3	<p>is player able to view the high score after click HIGHSCORE button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p>
4	<p>is player able to quit the game after click QUIT button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p>
Section 2 : FAT Level 1	
5	<p>Is the player able to move when press WASD / arrows button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p>

Table 4.1 Continued

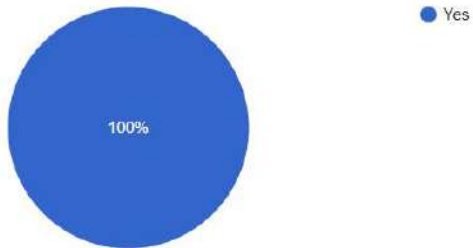
No	Question & Result				
6	<p>Is the player able to open the door when press E button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 6. The chart is a solid blue circle with the text '100%' centered inside. To the right of the chart is a legend with a blue dot and the text 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
7	<p>Is the player able to open the furniture / read message / talk to NPC when press SPACE button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 7. The chart is a solid blue circle with the text '100%' centered inside. To the right of the chart is a legend with a blue dot and the text 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
8	<p>Is the player able to speed up when press R and AD / left right arrow button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 8. The chart is a solid blue circle with the text '100%' centered inside. To the right of the chart is a legend with a blue dot and the text 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
9	<p>Is the player able to open the pause menu when click the pause icon button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 9. The chart is a solid blue circle with the text '100%' centered inside. To the right of the chart is a legend with a blue dot and the text 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				

Table 4.1 Continued

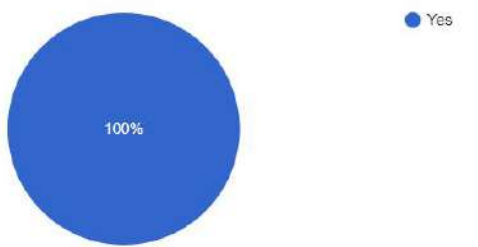
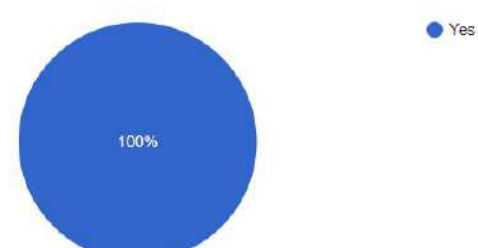

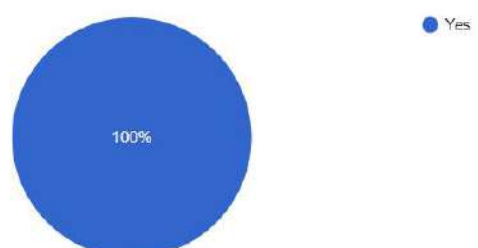
No	Question & Result
10	<p>Is the player able to shoot the enemy when press Q button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
11	<p>Is enemy able to be defeat and its health bar decrease when the bullet collide with it? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
12	<p>Is the lose animation played when player click YES button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
13	<p>Is player able to replay after click the REPLAY button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>

Table 4.1 Continued

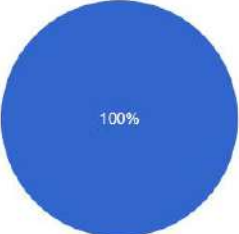
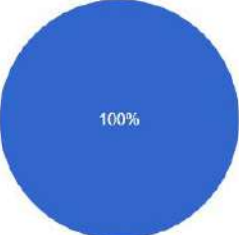
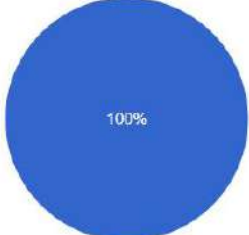

No	Question & Result		
14	<p>Is player's health bar decrease when attack by enemy? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><td>Yes</td><td>100%</td></tr></table>	Yes	100%
Yes	100%		
15	<p>Is player's health bar increase when collect the bandage? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><td>Yes</td><td>100%</td></tr></table>	Yes	100%
Yes	100%		
16	<p>Is player able to go to Level 2? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><td>Yes</td><td>100%</td></tr></table>	Yes	100%
Yes	100%		
Section 3 : FAT Level 2			
17	<p>Is the player able to move when press WASD / arrows button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><td>Yes</td><td>100%</td></tr></table>	Yes	100%
Yes	100%		

Table 4.1 Continued

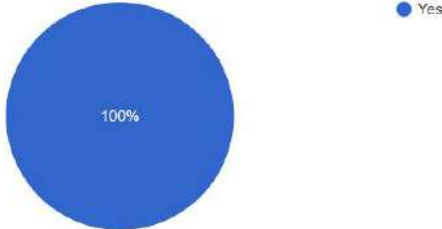



No	Question & Result
18	<p data-bbox="448 331 1182 387">Is the player able to open the door when press E button? (If no please state the problem in the 'other')</p> <p data-bbox="448 398 552 421">15 responses</p>  <p data-bbox="1027 461 1078 483">● Yes</p>
19	<p data-bbox="448 743 1198 799">Is the player able to open the furniture / read message / talk to NPC when press SPACE button? (If no please state the problem in the 'other')</p> <p data-bbox="448 810 552 833">15 responses</p>  <p data-bbox="1027 869 1078 891">● Yes</p>
20	<p data-bbox="448 1155 1305 1211">Is the player able to speed up when press R and AD / left right arrow button? (If no please state the problem in the 'other')</p> <p data-bbox="448 1223 552 1245">15 responses</p>  <p data-bbox="1098 1294 1149 1317">● Yes</p>
21	<p data-bbox="448 1590 1198 1646">Is the player able to open the pause menu when click the pause icon button? (If no please state the problem in the 'other')</p> <p data-bbox="448 1657 552 1680">15 responses</p>  <p data-bbox="1015 1720 1066 1742">● Yes</p>

Table 4.1 Continued

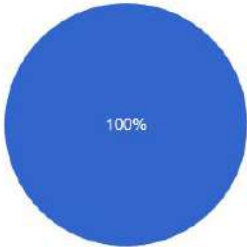



No	Question & Result
22	<p>Is the player able to shoot the enemy when press Q button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
23	<p>Is enemy able to be defeat and its health bar decrease when the bullet collide with it? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
24	<p>Is the lose animation played when player click YES button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
25	<p>Is the win animation played when player click correct answer? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>

Table 4.1 Continued

No	Question & Result
26	<p>Is player able to replay after click the REPLAY button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
27	<p>Is player's health bar decrease when attack by enemy? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
28	<p>Is player's health bar increase when collect the bandage? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
29	<p>Is background music / sound effect played correctly? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>

Table 4.1 Continued


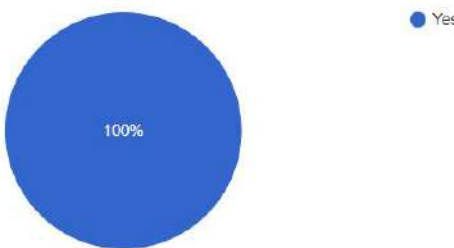
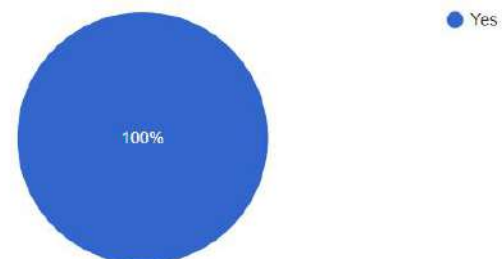
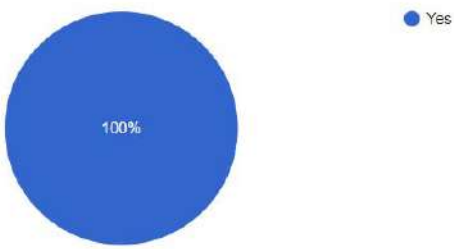
No	Question & Result
30	<p>Is player able to go to Level 3? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p>
<p>Section 4 : FAT Level 3</p>	
31	<p>Is the player able to move when press WASD / arrows button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p>
32	<p>Is the player able to open the door when press E button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p>
33	<p>Is the player able to open the furniture / read message / talk to NPC when press SPACE button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p>

Table 4.1 Continued



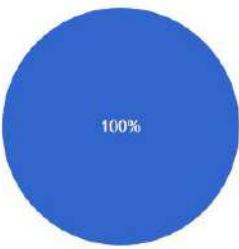

No	Question & Result
34	<p>Is the player able to speed up when press R and AD / left right arrow button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
35	<p>Is the player able to open the pause menu when click the pause icon button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
36	<p>Is the player able to shoot the enemy when press Q button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
37	<p>Is enemy able to be defeat and its health bar decrease when the bullet collide with it? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>

Table 4.1 Continued

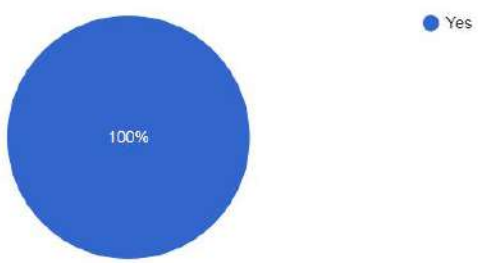
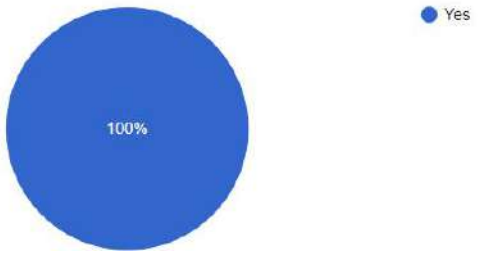
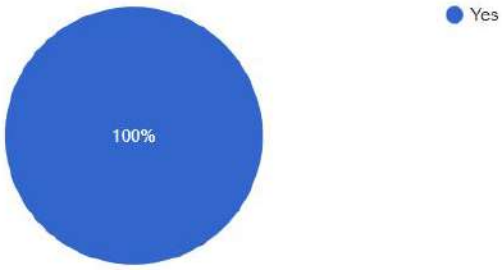
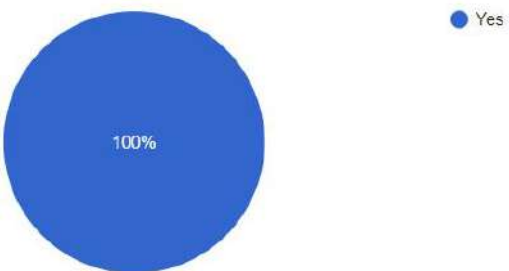
No	Question & Result				
38	<p>Is the lose animation played when player click wrong answer? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
39	<p>Is the win animation played when player click correct answer? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
40	<p>Is player able to replay after click the REPLAY button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
41	<p>Is player's health bar decrease when attack by enemy? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				

Table 4.1 Continued

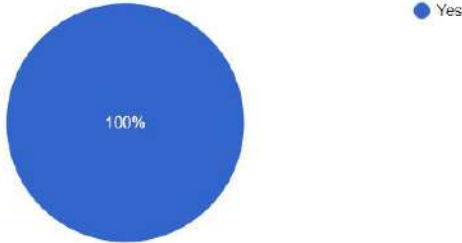

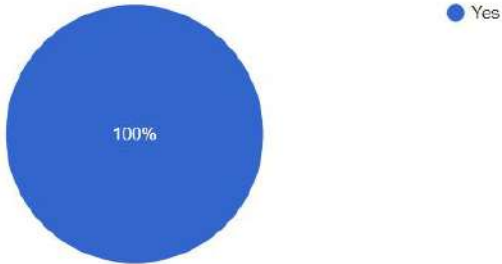
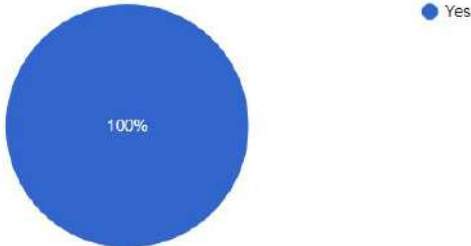
No	Question & Result				
42	<p>Is player's health bar increase when collect the bandage? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><th>Response</th><th>Percentage</th></tr><tr><td>Yes</td><td>100%</td></tr></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
43	<p>Is background music / sound effect played correctly? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><th>Response</th><th>Percentage</th></tr><tr><td>Yes</td><td>100%</td></tr></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
44	<p>Is player able to go to Level 4? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><th>Response</th><th>Percentage</th></tr><tr><td>Yes</td><td>100%</td></tr></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
Section 5 : Level 4					
45	<p>Is the player able to move when press WASD / arrows button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100%. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><tr><th>Response</th><th>Percentage</th></tr><tr><td>Yes</td><td>100%</td></tr></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				

Table 4.1 Continued

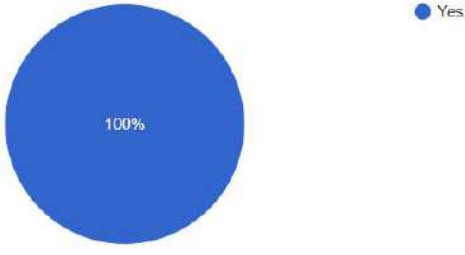
No	Question & Result				
46	<p>Is the player able to open the door when press E button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
47	<p>Is the player able to open the furniture / read message / talk to NPC when press SPACE button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
48	<p>Is the player able to speed up when press R and AD / left right arrow button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				
49	<p>Is the player able to open the pause menu when click the pause icon button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue dot next to the word 'Yes'.</p> <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>Yes</td><td>100%</td></tr></tbody></table>	Response	Percentage	Yes	100%
Response	Percentage				
Yes	100%				

Table 4.1 Continued

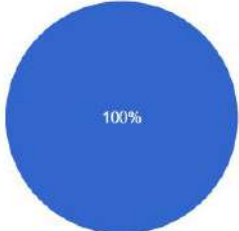
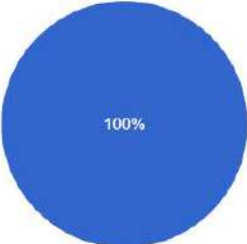
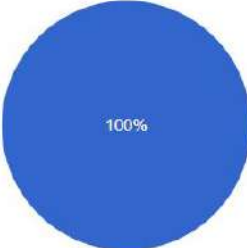
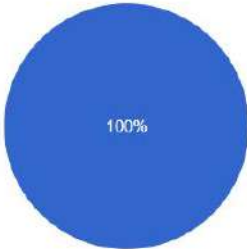
No	Question & Result
50	<p>Is the player able to shoot the enemy when press Q button? (if no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
51	<p>Is enemy able to be defeat and its health bar decrease when the bullet collide with it? (if no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
52	<p>Is the lose animation played when player click wrong answer? (if no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>
53	<p>Is the win animation played when player click correct answer? (if no please state the problem in the 'other')</p> <p>15 responses</p>  <p>100%</p> <p>● Yes</p>

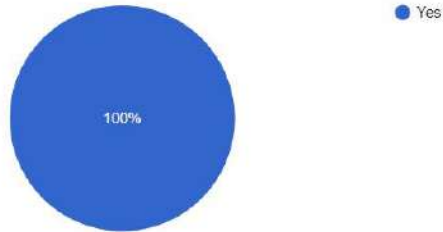
Table 4.1 Continued

No	Question & Result
54	<p>Is player able to replay after click the REPLAY button? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 54. The chart is a solid blue circle with the text "100%" in the center. To the right of the chart is a legend with a blue dot and the text "Yes".</p>
55	<p>Is player's health bar decrease when attack by enemy? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 55. The chart is a solid blue circle with the text "100%" in the center. To the right of the chart is a legend with a blue dot and the text "Yes".</p>
56	<p>Is player's health bar increase when collect the bandage? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 56. The chart is a solid blue circle with the text "100%" in the center. To the right of the chart is a legend with a blue dot and the text "Yes".</p>
57	<p>Is background music / sound effect played correctly? (If no please state the problem in the 'other')</p> <p>15 responses</p>  <p>A pie chart representing 100% of responses for question 57. The chart is a solid blue circle with the text "100%" in the center. To the right of the chart is a legend with a blue dot and the text "Yes".</p>

Table 4.1 Continued

No	Question & Result
----	-------------------

58 Is player able to go to Win Scene? (If no please state the problem in the 'other')
15 responses

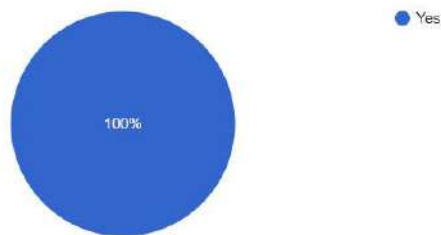


Section 6 : FAT Win Scene

59 Is player able to view the score, new high score and high score? (If no please state the problem in the 'other')
15 responses



60 Is player able to go to Main Menu after click the MAIN MENU button? (If no please state the problem in the 'other')
15 responses



Section 7 : UAT

61 Is Found You able to run on PC?
15 responses

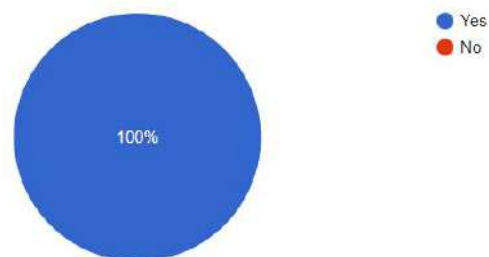


Table 4.1 Continued

No	Question & Result												
62	<p>Is Found You interesting? Copy</p> <p>15 responses</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0%</td> </tr> <tr> <td>2</td> <td>2</td> <td>13.3%</td> </tr> <tr> <td>3</td> <td>13</td> <td>86.7%</td> </tr> </tbody> </table>	Response	Count	Percentage	1	0	0%	2	2	13.3%	3	13	86.7%
Response	Count	Percentage											
1	0	0%											
2	2	13.3%											
3	13	86.7%											
63	<p>Is Found You easy to understand? Copy</p> <p>15 responses</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0%</td> </tr> <tr> <td>2</td> <td>2</td> <td>13.3%</td> </tr> <tr> <td>3</td> <td>13</td> <td>86.7%</td> </tr> </tbody> </table>	Response	Count	Percentage	1	0	0%	2	2	13.3%	3	13	86.7%
Response	Count	Percentage											
1	0	0%											
2	2	13.3%											
3	13	86.7%											
64	<p>Is Found You able to teach player about the techniques of protect themselves from being kidnapped? Copy</p> <p>15 responses</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0%</td> </tr> <tr> <td>2</td> <td>0</td> <td>0%</td> </tr> <tr> <td>3</td> <td>15</td> <td>100%</td> </tr> </tbody> </table>	Response	Count	Percentage	1	0	0%	2	0	0%	3	15	100%
Response	Count	Percentage											
1	0	0%											
2	0	0%											
3	15	100%											
65	<p>Is Found You able to let players know about the techniques / treats that will be used by kidnapper? Copy</p> <p>15 responses</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0%</td> </tr> <tr> <td>2</td> <td>0</td> <td>0%</td> </tr> <tr> <td>3</td> <td>15</td> <td>100%</td> </tr> </tbody> </table>	Response	Count	Percentage	1	0	0%	2	0	0%	3	15	100%
Response	Count	Percentage											
1	0	0%											
2	0	0%											
3	15	100%											

Table 4.1 Continued


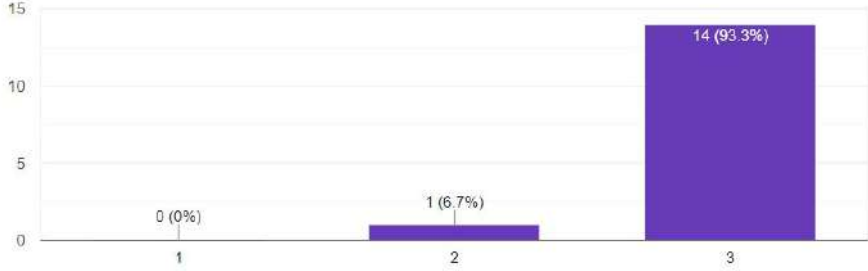
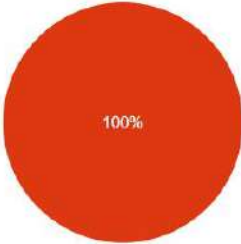
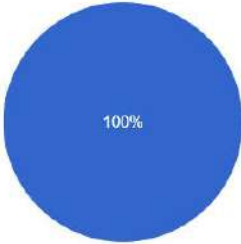
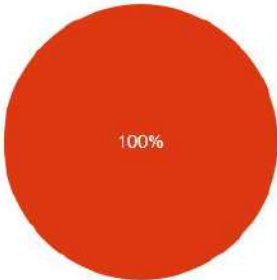
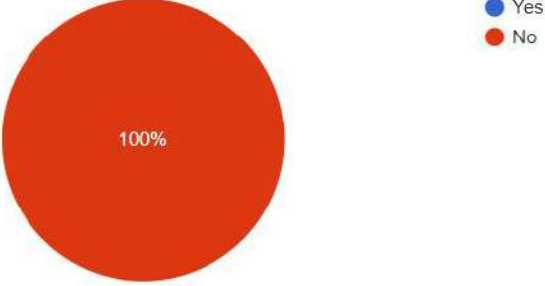
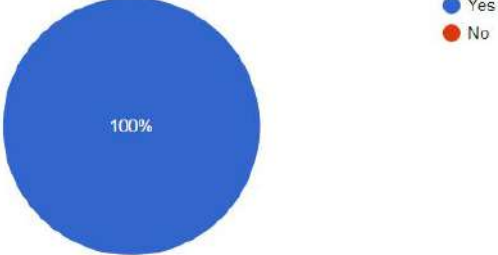

No	Question & Result												
<p>66</p>	<p>Is Found You able to increase the awareness of child abduction in player?  Copy</p> <p>15 responses</p>  <table border="1"> <caption>Awareness Levels</caption> <thead> <tr> <th>Level</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0%</td> </tr> <tr> <td>2</td> <td>1</td> <td>6.7%</td> </tr> <tr> <td>3</td> <td>14</td> <td>93.3%</td> </tr> </tbody> </table>	Level	Count	Percentage	1	0	0%	2	1	6.7%	3	14	93.3%
Level	Count	Percentage											
1	0	0%											
2	1	6.7%											
3	14	93.3%											
<p>67</p>	<p>In level 1, when a stranger ask to open the door for him. Will open the door?</p> <p>15 responses</p>  <ul style="list-style-type: none"> ● Yes ● No ● I don't know the answer 												
<p>68</p>	<p>In level 2, when you harassed by a stranger. What we can do?</p> <p>15 responses</p>  <ul style="list-style-type: none"> ● Seek adult help ● Do nothing ● I don't know the answer 												
<p>69</p>	<p>In level 2, should we walk alone in back alley / less crowded place?</p> <p>15 responses</p>  <ul style="list-style-type: none"> ● Yes ● No 												

Table 4.1 Continued

No	Question & Result
70	<p>In level 3, should we accept a ride / treats from a stranger?</p> <p>15 responses</p>  <p>A pie chart with a single red segment representing 100% of the responses. A legend to the right shows a blue circle for 'Yes' and a red circle for 'No'.</p>
71	<p>In level 4, should we report to police if we know about a kidnapping?</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows a blue circle for 'Yes' and a red circle for 'No'.</p>
72	<p>In level 4, what we should do if caught by kidnapper?</p> <p>15 responses</p>  <p>A pie chart with a single blue segment representing 100% of the responses. A legend to the right shows three options: a blue circle for 'Try anything (kick/bite) to escape from kidnapper', a red circle for 'Stay quiet', and a yellow circle for 'I don't know the answer'.</p>

4.4 Chapter Summary

This chapter is to explain the Design, Development and Evaluation process of the project. The game assets such as game objects, characters, NPC and Background Image is drawn using Adobe Photoshop. For the Development process, the game scripting is done by using Microsoft Visual Studio 2017 and game development is done by using Unity Engine. According to the result of the FAT and UAT form, the project is working well and the project is meet the purpose and objectives.

CHAPTER 5

CONCLUSION

5.1 Introduction

Game-based child abduction awareness can help the public increase awareness and understanding of child abduction through entertainment. The game application is focused on the techniques can be used to avoid the kidnappers and the techniques will used by kidnapper to kidnap.

The first objective is to study the existing child abduction awareness game application. The first objective achieved by searching 3 similar existing game applications which is a 2D game with theme of find and safe someone. The 3 existing games are Toby: The Secret Mine, Limbo, and the Final Station. The existing games are study on its user interface, functionality and the game design. Besides, a comparison table is create to compare the 3 existing games based on target player, game engine, platform, genre, operating system, processor, features, advantages, disadvantages and etc.

The second objective is to design and develop a 2D child abduction awareness game application. To achieve the objective, the game assets have been drawn using Adobe Photoshop to consistent the design of the game assets. Then, the game is developed using Unity Engine, where the level design is create in the software. After that, the scripting part is done in Microsoft Visual Studio 2017 which allow me to code all the needed script in C#. Once finish the development, the game application is export as .exe file for PC platform.

The third objective is to evaluate the functionality of the developed child Abduction awareness game application. To achieve the objective, the FAT and UAT have been created and then tested by 15 testers. The FAT form is used to test the function of the game application, while the UAT is used to check whether the game application meets the tester's player experience.

5.2 Limitation

There are several limitation and difficulty during design and development phase. The first limitation is the lack of the time. To ensure the game assets style is consistent, all the game assets and animation is drawn using Adobe Photoshop. The game assets creation had use a lot time to complete it. Besides, a lot of time is used in finding tutorial on the Internet, scripting, find and solving bugs and testing the result. Therefore, the time management is significant for a developer to finish the project on time.

Next, lack of tutorial also one of the limitation in the project development. Once the tutorial is fails to apply or cannot find a tutorial for a development, developer have to redo the design or find another alternative way to replace it and overcome the problem.

Further more, the lack of knowledge of the used software and scripting also affect on development process. Developer that lack of the knowledge need more time to find the solution and learn from the mistake.

5.3 Future Work

There are some feedbacks from testers to improve the project. First is the story of the game can be more smooth and interesting, because several testers feel the story not logic and the ending can be inprove. Next, testers suggest to add more interactable NPCs to make the game more fun.

REFERENCES

- 30+ games like LIMBO - SteamPeek. (2022, 6 April). STEAMPEEK. Retrieved March 21, 2022, from <https://steampeek.hu/?appid=48000#gdetails>
- 30+ games like The Final Station - SteamPeek. (2022, 6 April). STEAMPEEK. Retrieved March 22, 2022, from <https://steampeek.hu/?appid=435530>
- 30+ games like Toby: The Secret Mine - SteamPeek. (2022, 9 April). STEAMPEEK. Retrieved March 20, 2022, from <https://steampeek.hu/?appid=395160>
- Contributor, T. (2019, 15 August). *spiral model*. SearchSoftwareQuality. Retrieved April 13, 2022, from <https://www.techtarget.com/searchsoftwarequality/definition/spiral-model>
- Davey, L. (2021, 12 October). *How to thoroughly document your project requirements*. Teamwork. Retrieved April 13, 2022, from <https://www.teamwork.com/blog/project-requirements/>
- Dimovsky, A. (2022, February 15). 40+ Horrific & Chilling Kidnapping Statistics [2022] . Safe at Last. Retrieved April 14, 2022, from <https://safeatlast.co/blog/kidnapping-statistics/#gref>
- E. (2013, 23 October). *Limbo: The Pros and Cons and Confusions*. Narrative and Digital Technology. Retrieved March 22, 2022, from

<https://narrativedigital.wordpress.com/2013/10/22/limbo-the-pros-and-cons-and-confusions/>

E. (2013b, October 23). *Limbo: The Pros and Cons and Confusions*. Narrative and Digital Technology. Retrieved March 22, 2022, from <https://narrativedigital.wordpress.com/2013/10/22/limbo-the-pros-and-cons-and-confusions/>

Game Creation Guidelines/Spiral Model of Game Development - Wikiversity. (z.d.). Wikiversity. Retrieved April 13, 2022, from https://en.wikiversity.org/wiki/Game_Creation_Guidelines/Spiral_Model_of_Game_Development#:~:text=The%20Spiral%20Model%20is%20an,is%20thus%20an%20evolutionary%20process

Game Development Lifecycle Models | Studytonight. (z.d.). Study Tonight. Retrieved April 13, 2022, from <https://www.studytonight.com/3d-game-engineering-with-unity/game-development-models>

H. (2019, 19 juli). *Limbo PC Game Free Download*. FreeGamesLand | Full PC Games Free Download. Retrieved March 21, 2022, from <https://freegamesland.net/limbo-pc-game-free-download/>

H. (2019, 7 November). *Toby The Secret Mine Free Download*. FreeGamesLand | Full PC Games Free Download. Retrieved March 20, 2022, from <https://freegamesland.net/toby-the-secret-mine-free-download/>

- H. (2019a, May 15). *The Final Station Free PC Download*. FreeGamesLand | Full PC Games Free Download. Retrieved March 22, 2022, from <https://freegamesland.net/the-final-station-free-pc-download/>
- Houghton, D. (2010, 19 July). *Limbo review*. Gamesradar. Retrieved March 22, 2022, from <https://www.gamesradar.com/limbo-review/>
- Kinsey, D. J. (2017, 10 June). *Toby: The Secret Mine Review*. The Nerd Stash. Retrieved March 22, 2022, from <https://thenerdstash.com/toby-the-secret-mine-review/#:%7E:text=Toby%3A%20The%20Secret%20Mine%20is,several%20hours%20of%20challenging%20gameplay.&text=Pros%3A,Great%20puzzle%20balance>
- Lukas Navratil. (2015, 20 October). *Toby: The Secret Mine*. IGDB.Com. Retrieved March 20, 2022, from <https://www.igdb.com/games/toby-the-secret-mine>
- M. (2017, 7 August). *Spiral model for game development: Techniques to develop games*. Game Developer Tips. Retrieved April 13, 2022, from <https://gamedevelopertips.com/spiral-model-for-game-development/>
- Missing children's statistics*. Global Missing Children's Network. (2017, June 29). Retrieved March 16, 2022, from <https://globalmissingkids.org/awareness/missing-children-statistics/>
- Non-Functional Requirements | TankWars Project | Assembla*. (z.d.). "Assembla"/. Retrieved April 14, from https://app.assembla.com/wiki/show/tank_wars/Non_Functional_Requirement

Techopedia. (2011, 7 October). *Playtesting*. Techopedia.Com. Retrieved April 13, 2022, from <https://www.techopedia.com/definition/27197/playtesting>

The Final Station Review - Review. Nintendo World Report. (n.d.). Retrieved April 13, 2022, from <http://www.nintendoworldreport.com/review/46616/the-final-station-switch-review>

The Final Station. (2019, 6 March). TechSpot. Retrieved March 22, 2022, from <https://www.techspot.com/products/pc-games/the-final-station.149750/>

Vignes. (2021, February 6). *4 children go missing every day in Malaysia, here's what you can do about it...* Asklegal. Retrieved March 20, 2022, from <https://asklegal.my/p/child-missing-Malaysia-nur-alerts-mms-warning-signup>

What are the functional and non-functional requirements of a snake game? (z.d.). Quora. Retrieved April 14, from <https://www.quora.com/What-are-the-functional-and-non-functional-requirements-of-a-snake-game>

What is the Software Development Lifecycle? (2021, 12 August). Definition and Overview. Retrieved April 13, 2022, from <https://www.productplan.com/learn/software-development-lifecycle/#:%7E:text=The%20software%20development%20lifecycle%20>

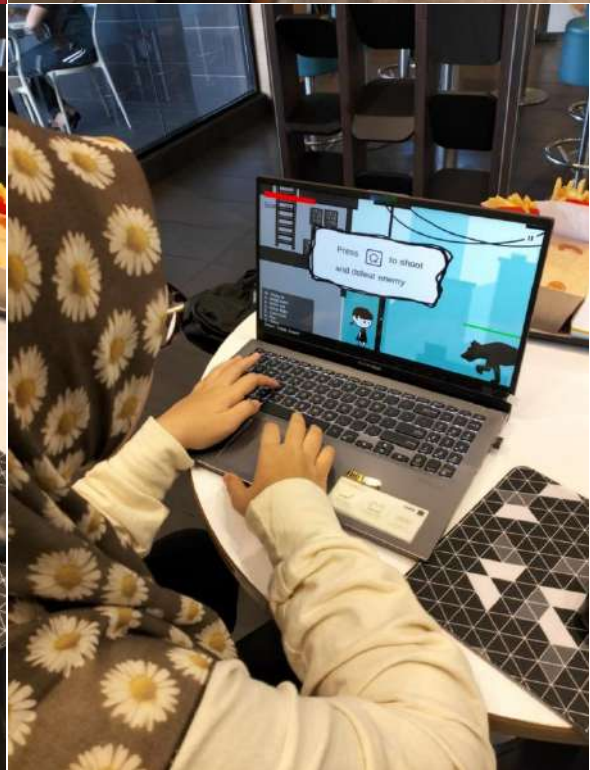
Wikipedia contributors. (2022, 16 February). *The Final Station*. Wikipedia. Retrieved March 22, 2022, from https://en.wikipedia.org/wiki/The_Final_Station

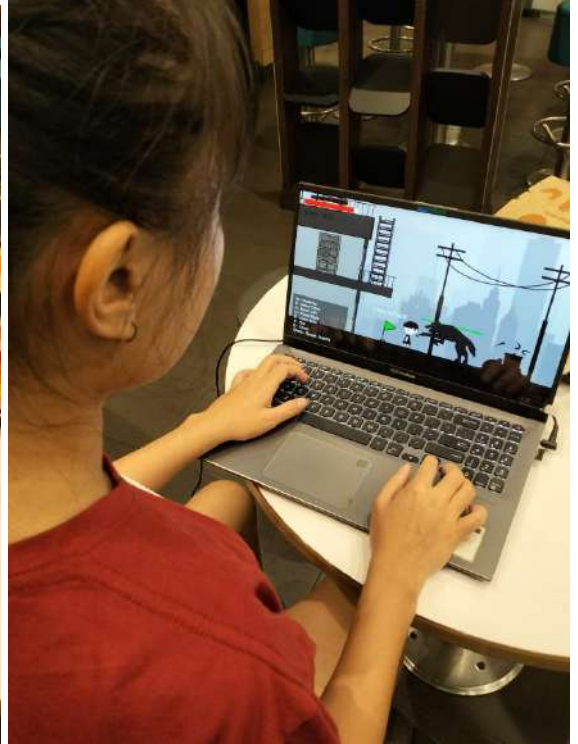
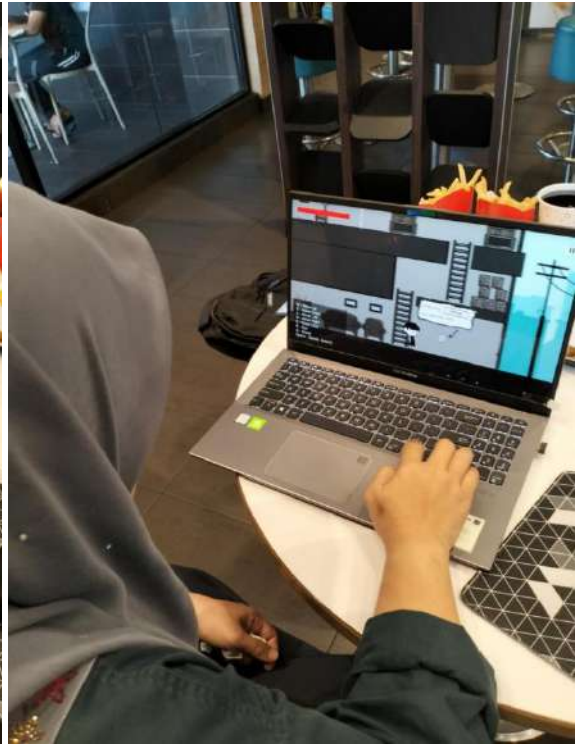
Wikipedia contributors. (2022a, January 8). *Toby: The Secret Mine*. Wikipedia. Retrieved March 20, 2022, from https://en.wikipedia.org/wiki/Toby:_The_Secret_Mine

Wikipedia contributors. (2022e, April 8). *Limbo (video game)*. Wikipedia. Retrieved March 21, 2022, from [https://en.wikipedia.org/wiki/Limbo_\(video_game\)](https://en.wikipedia.org/wiki/Limbo_(video_game))

APPENDICES

APPENDIX A - FAT & UAT Proof





APPENDIX A - C# Scripting

BulletSetting.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class bulletSetting : MonoBehaviour
{
    public float speed;
    Rigidbody2D rb;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();

        Transform player = GameObject.Find("Player").transform;
        rb.velocity = transform.right * player.localScale.x * speed;

        Destroy(gameObject, 1f);
    }
}
```

CameraFollow.cs

```
using UnityEngine;

public class CameraFollow : MonoBehaviour
{
    public Transform target;
    public Vector3 offset;

    private void FixedUpdate()
    {
        transform.position = target.position + offset;
    }
}
```

changeBackgroundMusic.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class changeBackgroundMusic : MonoBehaviour
{

```



```

public AudioSource lv4backgroundMusic;
public AudioSource jungleBackgroundMusic;

// Start is called before the first frame update
void Start()
{

}

private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.CompareTag("Player"))
    {
        lv4backgroundMusic.enabled = false;
        jungleBackgroundMusic.enabled = true;
    }
}

private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.gameObject.CompareTag("Player"))
    {
        lv4backgroundMusic.enabled = true;
        jungleBackgroundMusic.enabled = false;
    }
}
}

```

ChangeScene.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class ChangeScene : MonoBehaviour
{
    public GameObject panel;
    public GameObject emilyShoePanel;
    public GameObject emilyShoe;
    public GameObject emilyShoeInstruction;
    public GameObject TimelineGameBegin;
    public GameObject losepanel;

    private int currentSceneIndex;
    private int sceneToContinue;
}

```

```

public AudioSource clickSound;

public bool open = false;

private void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        clickSound.Play();
    }

    if (open)
    {
        Time.timeScale = 0;
    }
    else
    {
        Time.timeScale = 1;
    }
}

public void NewGame()
{
    SceneManager.LoadScene(1);
    open = false;
    PlayerPrefs.DeleteKey("savedScore1");
    PlayerPrefs.DeleteKey("savedScore2");
    PlayerPrefs.DeleteKey("savedScore3");
    PlayerPrefs.DeleteKey("savedScore4");

    PlayerPrefs.DeleteKey("healthlv1");
    PlayerPrefs.DeleteKey("healthlv2");
    PlayerPrefs.DeleteKey("healthlv3");
    PlayerPrefs.DeleteKey("healthlv4");
}

public void loadScene(int index)
{
    clickSound.Play();

    SceneManager.LoadScene(index);
    open = false;
}

public void exit()
{
    Application.Quit();
    Debug.Log("Game is exiting");
}

```

```

public void openPause()
{
    clickSound.Play();
    panel.SetActive(true);
    open = true;
}

public void closePause()
{
    clickSound.Play();

    panel.SetActive(false);
    open = false;
}

public void pause_exit()
{
    clickSound.Play();

    currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
    PlayerPrefs.SetInt("SavedScene", currentSceneIndex);
    SceneManager.LoadScene(0);
    open = false;
}

public void continueBtn()
{
    clickSound.Play();

    sceneToContinue = PlayerPrefs.GetInt("SavedScene");
    open = false;
    if(sceneToContinue != 0)
    {
        SceneManager.LoadScene(sceneToContinue);
        Time.timeScale = 1;
    }
    else
    {
        return;
    }
}

public void emilyShoeClosebtn()
{
    clickSound.Play();

    open = false;
    emilyShoePanel.SetActive(false);
    emilyShoe.SetActive(false);
    emilyShoeInstruction.SetActive(true);
}

```

```
        Destroy(emilyShoeInstruction, 3f);
    }

}
```

ChangeScene_highscore.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class ChangeScene_highscore : MonoBehaviour
{
    public GameObject panel;
    public GameObject emilyShoePanel;
    public GameObject emilyShoe;
    public GameObject emilyShoeInstruction;
    public GameObject TimelineGameBegin;
    public GameObject losepanel;

    public GameObject mainMenuUI;
    public GameObject highScoreUI;

    public Text score;
    public Text newhighscore;
    public Text highscore;

    private int currentSceneIndex;
    private int sceneToContinue;

    public AudioSource clickSound;

    public bool open = false;
    public bool isNewGame = false;

    public float timer;

    private void Start()
    {
        score.text = "SCORE : " + PlayerPrefs.GetInt("currentscore");
        newhighscore.text = "NEW HIGH SCORE : " +
PlayerPrefs.GetInt("newhighscore");
        highscore.text = "HIGH SCORE : " + PlayerPrefs.GetInt("oldhighscore");
    }
}
```

```

}

private void Update()
{
    if(Input.GetMouseButtonDown(0))
    {
        clickSound.Play();
    }

    if(isNewGame)
    {
        timer -= Time.deltaTime;

        if(timer <= 0)
        {
            SceneManager.LoadScene(7);
            open = false;
            PlayerPrefs.DeleteKey("savedScore1");
            PlayerPrefs.DeleteKey("savedScore2");
            PlayerPrefs.DeleteKey("savedScore3");
            PlayerPrefs.DeleteKey("savedScore4");

            PlayerPrefs.DeleteKey("healthlv1");
            PlayerPrefs.DeleteKey("healthlv2");
            PlayerPrefs.DeleteKey("healthlv3");
            PlayerPrefs.DeleteKey("healthlv4");
            isNewGame = false;
        }
    }

    if(open)
    {
        Time.timeScale = 0;
    }
    else
    {
        Time.timeScale = 1;
    }
}

public void NewGame()
{
    clickSound.Play();
    isNewGame = true;
}

public void loadScene(int index)
{
    clickSound.Play();
}

```

```

    SceneManager.LoadScene(index);
    open = false;
}

public void exit()
{
    clickSound.Play();

    Application.Quit();
    Debug.Log("Game is exiting");
}

public void openPause()
{
    clickSound.Play();

    panel.SetActive(true);
    open = true;
}

public void closePause()
{
    clickSound.Play();

    panel.SetActive(false);
    open = false;
}

public void pause_exit()
{
    clickSound.Play();

    currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
    PlayerPrefs.SetInt("SavedScene", currentSceneIndex);
    SceneManager.LoadScene(0);
    open = false;
}

public void continueBtn()
{
    clickSound.Play();

    sceneToContinue = PlayerPrefs.GetInt("SavedScene");
    open = false;

    if (sceneToContinue != 0)
    {
        SceneManager.LoadScene(sceneToContinue);
        Time.timeScale = 1;
    }
}

```

```

    }
    else
    {
        return;
    }
}

public void emilyShoeClosebtn()
{
    clickSound.Play();

    open = false;
    emilyShoePanel.SetActive(false);
    emilyShoe.SetActive(false);
    emilyShoeInstruction.SetActive(true);
    Destroy(emilyShoeInstruction, 3f);
}

public void deleteData()
{
    clickSound.Play();

    PlayerPrefs.DeleteKey("oldhighscore");
    PlayerPrefs.DeleteKey("newhighscore");
    PlayerPrefs.DeleteKey("currentscore");
    PlayerPrefs.DeleteKey("highscore");

    score.text = "SCORE : " + PlayerPrefs.GetInt("currentscore");
    newhighscore.text = "NEW HIGH SCORE : " +
PlayerPrefs.GetInt("newhighscore");
    highscore.text = "HIGH SCORE : " + PlayerPrefs.GetInt("oldhighscore");
}

public void HighScoreBtn()
{
    clickSound.Play();

    mainMenuUI.SetActive(false);
    highScoreUI.SetActive(true);
}

public void HighScoreBackBtn()
{
    clickSound.Play();

    mainMenuUI.SetActive(true);
    highScoreUI.SetActive(false);
}

```

```
}
```

checkpoint.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class checkpoint : MonoBehaviour
{
    public int index;

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if(collision.gameObject.CompareTag("Player"))
        {
            SceneManager.LoadScene(index);
        }
    }
}
```

checkpointPlayerlv2.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Playables;

public class checkpointPlayerlv2 : MonoBehaviour
{
    public Vector3 respawnPoint;
    public Vector3 KidnapperPos;
    public Transform kidenaperTransform;
    public GameObject losePanel;
    public PlayerController playerControlScript;

    public BoxCollider2D theCheckpointCollider;
    public GameObject helpDetector;
    public BoxCollider2D kidnapperChaseDetectorCollider;
    public GameObject closeDoorDetector;
    public lv2KidnapperFollowPlayer kidnapperChaseScript;
```



```

public GameObject kidanpperChaseTimeline;
public Image fade;
public CameraFollow camaraScript;
public BoxCollider2D closeDoorDetectorCollider;
public GameObject door;

public AudioSource kidnapperTheme;

// Start is called before the first frame update
void Start()
{
    respawnPoint = transform.position;
    KidnapperPos = kidenaperTransform.position;
}

// Update is called once per frame
void Update()
{

}

private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.CompareTag("checkpointMini"))
    {
        respawnPoint = transform.position;
        theCheckpointCollider.enabled = false;
    }
}

public void respawn()
{
    transform.position = respawnPoint;
    kidenaperTransform.position = KidnapperPos;
    fade.transform.localScale = new Vector3(0, 0, 0);
    playerControlScript.enabled = true;

    helpDetector.SetActive(false);
    kidnapperChaseDetectorCollider.enabled = true;
    closeDoorDetector.SetActive(false);
    kidnapperChaseScript.enabled = false;
    kidanpperChaseTimeline.SetActive(true);
    losePanel.SetActive(false);
    camaraScript.enabled = true;
    closeDoorDetectorCollider.enabled = true;
    door.SetActive(false);
    kidnapperTheme.enabled = false;
}
}

```

checkpointPlayerlv3.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Playables;

public class checkpointPlayerlv3 : MonoBehaviour
{
    public Vector3 respawnPoint;
    public Vector3 KidnapperPos;
    public Vector3 vanPos;
    public Transform van;
    public Transform kidenaperTransform;
    public GameObject losePanel;
    public PlayerController playerControlScript;

    public BoxCollider2D theCheckpointCollider;
    public BoxCollider2D kidnapperDetectorCollider;
    public Image fade;
    public CameraFollow camaraScript;
    public Camera mainCam;

    public GameObject selectionPanel3_1;
    public GameObject selectionPanel3_2;
    public GameObject timelinePart2;
    public GameObject timelineloseAction;

    public float currentHealth;
    public Health3 playerHealthScript;
    public Image healthBar;

    // Start is called before the first frame update
    void Start()
    {
        respawnPoint = transform.position;
        KidnapperPos = kidenaperTransform.position;
        vanPos = van.position;
        currentHealth = healthBar.fillAmount;
    }

    // Update is called once per frame
    void Update()
    {
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
    }
}
```

```

        if(collision.gameObject.CompareTag("checkpointMini"))
        {
            respawnPoint = transform.position;
            KidnapperPos = kidnaperTransform.position;
            vanPos = van.position;
            currentHealth = healthBar.fillAmount;
            theCheckpointCollider.enabled = false;
        }
    }

    public void respawn()
    {
        transform.position = respawnPoint;
        kidnaperTransform.position = KidnapperPos;
        van.position = vanPos;
        transform.localScale = new Vector2(0.15f, 0.15f);
        kidnaperTransform.localScale = new Vector2(-0.09f, 0.08f);
        van.localScale = new Vector2(0.25f, 0.25f);
        kidnaperTransform.position = KidnapperPos;
        fade.transform.localScale = new Vector3(0, 0, 0);
        playerControlScript.enabled = true;

        losePanel.SetActive(false);
        camaraScript.enabled = true;
        mainCam.orthographicSize = 1.15f;

        selectionPanel3_2.SetActive(false);
        selectionPanel3_1.SetActive(false);
        timelinePart2.SetActive(true);
        timelineloseAction.SetActive(true);

        kidnapperDetectorCollider.enabled = true;
        playerHealthScript.healthAfterRespawn(currentHealth);
    }
}

```

checkpointPlayerlv4.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Playables;

public class checkpointPlayerlv4 : MonoBehaviour
{
    public Vector3 respawnPoint;
    public Vector3 KidnapperPos;
}

```

```

public Vector3 Kidnapper2Pos;
public Vector3 Kidnapper3Pos;
public Vector3 PolicePos;
public float currentHealth;

public Image healthBar;
public Transform kidenaperTransform;
public Transform kidenaper2Transform;
public Transform kidenaper3Transform;
public Transform policeTransform;
public GameObject losePanel;
public GameObject losePanel1;
public PlayerController playerControlScript;
public Shooting playerShootScript;

public BoxCollider2D theCheckpointCollider;
public BoxCollider2D theCheckpointCollider2;

public GameObject kidnapperTalkDetector;
public GameObject detector1;
public GameObject detector2;
public lv4KidnapperChase kidnapperChaseScript;
public lv4KidnapperChase kidnapperChaseScript2;
public Image fade;
public CameraFollow camaraScript;
public Camera mainCam;
public GameObject policeDialogue;
public GameObject firstKidnapper;
public GameObject secondKidnapper;
public GameObject thirdKidnapper;
public lv4KidnapperChase thirdKidnapperscript;
public GameObject struggle1;
public GameObject struggle2;
public GameObject struggle3;
public Health4 playerHealthScript;
public GameObject thedoor;
public GameObject AbandonedHousedoor;
public GameObject AbandonedHousedoor2;
public GameObject theSquare;

public GameObject kidnapperGroup;
public BoxCollider2D kidnapperTalkCollider;
public SpriteRenderer square;
public GameObject loseCollider;
public BoxCollider2D loseColliderCOL;

public GameObject endingLoseKidnapper1;
public GameObject endingLoseKidnapper2;
public AudioSource kidnaperTheme;
public GameObject policeDialogueScale;

```

```

public lv4SelectionScoreCounter thePoliceScoreScript;
public GameObject waypoint5;
public BoxCollider2D waypoint5Collider;
public lv4StopChase stopKidnapperScript;

public GameObject enemy1;
public GameObject enemy2;

private int num = 0;

// Start is called before the first frame update
void Start()
{
    respawnPoint = transform.position;
    KidnapperPos = kidnaperTransform.position;
    Kidnapper2Pos = kidnaper2Transform.position;
    Kidnapper3Pos = kidnaper3Transform.position;
    PolicePos = policeTransform.position;
    currentHealth = healthBar.fillAmount;
}

// Update is called once per frame
void Update()
{

}

private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.CompareTag("checkpointMini"))
    {
        currentHealth = healthBar.fillAmount;
        respawnPoint = transform.position;
        theCheckpointCollider.enabled = false;
    }
    else if(collision.gameObject.CompareTag("checkpointMini2"))
    {
        currentHealth = healthBar.fillAmount;
        respawnPoint = transform.position;
        theCheckpointCollider2.enabled = false;
    }
}

public void respawn()
{
    transform.position = respawnPoint;
    kidnaperTransform.position = KidnapperPos;
    kidnaper2Transform.position = Kidnapper2Pos;
}

```

```

kidenaper3Transform.position = Kidnapper3Pos;
kidenaper3Transform.localScale = new Vector2(0.09f, 0.09f);
policeTransform.position = PolicePos;
healthBar.fillAmount = currentHealth;
policeTransform.localScale = new Vector2(-0.1f, 0.1f);
fade.transform.localScale = new Vector3(0, 0, 0);
playerControlScript.enabled = true;
playerShootScript.enabled = true;

kidnapperTalkDetector.SetActive(true);
kidnapperChaseScript.enabled = false;
detector1.SetActive(false);
detector2.SetActive(false);
losePanel.SetActive(false);
losePanel1.SetActive(false);
camaraScript.enabled = true;
policeDialogue.SetActive(true);
firstKidnapper.SetActive(true);
secondKidnapper.SetActive(false);
thirdKidnapper.SetActive(false);
thirdKidnapperscript.enabled = true;
struggle1.SetActive(false);
struggle2.SetActive(false);
struggle3.SetActive(false);
thedoors.SetActive(true);
AbandonedHousedoor.SetActive(true);
AbandonedHousedoor2.transform.localScale = new Vector2(1.5f, 1);
theSquare.SetActive(true);

square.color = new Color(0, 0, 0, 255);
kidnapperTalkCollider.enabled = true;
kidnapperGroup.SetActive(true);
loseCollider.SetActive(false);
loseColliderCOL.enabled = true;

endingLoseKidnapper1.transform.localScale = new Vector2(0, 0);
endingLoseKidnapper2.transform.localScale = new Vector2(0, 0);
kidnaperTheme.enabled = false;

kidnapperChaseScript.respawnIndex(num);
kidnapperChaseScript2.respawnIndex(num);
thirdKidnapperscript.respawnIndex(num);
policeDialogueScale.transform.localScale = new Vector2(-10f, 10f);

thePoliceScoreScript.respawnPoliceScore(num);
waypoint5.SetActive(true);
waypoint5Collider.enabled = false;
stopKidnapperScript.enabled = false;

enemy1.SetActive(true);

```

```
        enemy2.SetActive(true);

        mainCam.orthographicSize = 1.15f;

        playerHealthScript.healthAfterRespawn(currentHealth);
    }
}
```

closeButton.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;

public class closeButton : MonoBehaviour
{
    public GameObject pwPanel;
    public GameObject playAnim;

    // Start is called before the first frame update
    void Start()
    {
        gameObject.GetComponent<Button>().onClick.AddListener(ButtonClicked);
    }

    private void ButtonClicked()
    {
        pwPanel.SetActive(false);
        playAnim.SetActive(true);
        Time.timeScale = 1;
    }
}
```

closeButtonMessage.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;

public class closeButtonMessage : MonoBehaviour
{
    public GameObject ThePanel;

    // Start is called before the first frame update
    void Start()
    {
        gameObject.GetComponent<Button>().onClick.AddListener(ButtonClicked);
    }

    private void ButtonClicked()
    {
        ThePanel.SetActive(false);
        Time.timeScale = 1;
    }
}
```

closeCluebtn.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;

public class closeCluebtn : MonoBehaviour
{
    public GameObject ThePanel;
    public FoundClue foundClueScript;
    bool yes = false;

    // Start is called before the first frame update
    void Start()
    {
        gameObject.GetComponent<Button>().onClick.AddListener(ButtonClicked);
    }
}
```



```

    }

    private void ButtonClicked()
    {
        ThePanel.SetActive(false);
        Time.timeScale = 1;

        yes = true;
        foundClueScript.showAnneSelfTalk(yes);
    }
}

```

collideBandageLv1.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideBandageLv1 : MonoBehaviour
{
    private Animator anim;
    private bool selected = false;
    public Health1 healthScript;
    public KeyCode keycode;

    public AudioSource healingSound;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (selected && Input.GetKeyDown(keycode))
        {
            healingSound.Play();
            gameObject.SetActive(false);
            healthScript.Heal(50);
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelect", true);
            selected = true;
        }
    }
}

```

```

    }
}

private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        anim.SetBool("isSelect", false);
        selected = false;
    }
}
}

```

collideBandageLv2.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideBandageLv2 : MonoBehaviour
{
    private Animator anim;
    private bool selected = false;
    public Health2 healthScript;
    public KeyCode keycode;

    public AudioSource healingSound;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (selected && Input.GetKeyDown(keycode))
        {
            healingSound.Play();
            gameObject.SetActive(false);
            healthScript.Heal(50);
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {

```

```

        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelect", true);
            selected = true;
        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelect", false);
            selected = false;
        }
    }
}

```

collideBandageLv3.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideBandageLv3 : MonoBehaviour
{
    private Animator anim;
    private bool selected = false;
    public Health3 healthScript;
    public KeyCode keycode;

    public AudioSource healingSound;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (selected && Input.GetKeyDown(keycode))
        {
            healingSound.Play();
            gameObject.SetActive(false);
            healthScript.Heal(50);
        }
    }
}

```

```

    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelected", true);
            selected = true;

        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelected", false);
            selected = false;
        }
    }
}

```

collideBandageLv4.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideBandageLv4 : MonoBehaviour
{
    private Animator anim;
    private bool selected = false;
    public Health4 healthScript;
    public KeyCode keycode;

    public AudioSource healingSound;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    void Update()
    {
        if (selected && Input.GetKeyDown(keycode))
        {

```

```

        healingSound.Play();
        gameObject.SetActive(false);
        healthScript.Heal(50);
    }
}

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        anim.SetBool("isSelected", true);
        selected = true;
    }
}

private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        anim.SetBool("isSelected", false);
        selected = false;
    }
}
}

```

collideMessage.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideMessage : MonoBehaviour
{
    private bool collided = false;
    public GameObject Panel;
    public KeyCode keycode;

    public AudioSource openMessageSound;

    void Start()
    {
    }
}

```

```

void Update()
{
    if (collided && Input.GetKeyDown(keycode))
    {
        openMessageSound.Play();
        Panel.SetActive(true);
    }
}

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        collided = true;
    }
}

private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.CompareTag("Player"))
    {
        collided = false;
    }
}
}

```

collideNotes.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class collideNotes : MonoBehaviour
{
    private Animator anim;
    private bool selected = false;
    public GameObject pwPanel;
    public KeyCode keycode;

    public AudioSource readMessageSound;

    void Start()
    {
        anim = GetComponent<Animator>();
    }
}

```

```

    }

    void Update()
    {
        if (selected && Input.GetKeyDown(keycode))
        {
            readMessageSound.Play();
            pwPanel.SetActive(true);
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelect", true);
            selected = true;
        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            anim.SetBool("isSelect", false);
            selected = false;
        }
    }
}

```

Dialogue.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class Dialogue : MonoBehaviour
{
    public GameObject window;
    public GameObject indicator;
    public List<string> dialogues;
    public TMP_Text dialogueText;
    public float writingSpeed;

    private int index;

```

```

private int charIndex;

private bool started;
private bool waitForNext;

private void Awake()
{
    ToggleIndicator(false);
    ToggleWindow(false);
}

public void ToggleWindow(bool show)
{
    window.SetActive(show);
}

public void ToggleIndicator(bool show)
{
    indicator.SetActive(show);
}

public void StartDialogue()
{
    if (started)
        return;

    started = true;
    ToggleWindow(true);
    ToggleIndicator(false);
    GetDialogue(0);
}

private void GetDialogue(int i)
{
    index = i;
    charIndex = 0;
    dialogueText.text = string.Empty;
    StartCoroutine(Writing());
}

public void EndDialogue()
{
    started = false;
    waitForNext = false;
    StopAllCoroutines();
    ToggleWindow(false);
}

IEnumerator Writing()
{

```



```

yield return new WaitForSeconds(writingSpeed);

string currentDialogue = dialogues[index];
dialogueText.text += currentDialogue[charIndex];
charIndex++;

if(charIndex < currentDialogue.Length)
{
yield return new WaitForSeconds(writingSpeed);
StartCoroutine(Writing());
}
else
{
waitForNext = true;
}
}

private void Update()
{
if (!started)
return;
if (waitForNext)
{
waitForNext = false;
index++;

StartCoroutine(delay());
}
}

IEnumerator delay()
{
yield return new WaitForSeconds(2f);

if (index < dialogues.Count)
{
GetDialogue(index);
}
else
{
ToggleIndicator(true);
EndDialogue();
}
}
}

```

DialogueTrigger.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DialogueTrigger : MonoBehaviour
{
    private bool playerDetected;
    public Dialogue dialogueScript;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.tag == "Player")
        {
            playerDetected = true;
            dialogueScript.ToggleIndicator(playerDetected);
        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.tag == "Player")
        {
            playerDetected = false;
            dialogueScript.ToggleIndicator(playerDetected);
            dialogueScript.EndDialogue();
        }
    }

    private void Update()
    {
        if(playerDetected && Input.GetKeyDown(KeyCode.Space))
        {
            dialogueScript.StartDialogue();
        }
    }
}
```