

ENVIRONMENTAL SUSTAINABILITY GAME
FOR ELEMENTARY SCHOOL STUDENTS
(GREEN GENIE)

GOH MEI KEI

BACHELOR OF COMPUTER SCIENCE
(GRAPHICS & MULTIMEDIA TECHNOLOGY)
WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : GOH MEI KEI

Date of Birth : _____

Title : ENVIRONMENTAL SUSTAINABILITY GAME FOR
ELEMENTARY SCHOOL STUDENTS (GREEN GENIE)

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)

GOH MEI KEI

Dr. Danakorn Nincarean A/L Eh Phon

Date: 2/13/2023

Date: 13/2/2023

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

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

 (ii)

 (iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in

(Supervisor's Signature)

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

(Co-supervisor's Signature)

Full Name :
Position :
Date :



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.

Mei Kei

(Student's Signature)

Full Name : GOH MEI KEI

ID Number : CD19047

Date : 13 February 2023

ENVIRONMENTAL SUSTAINABILITY GAME
FOR ELEMENTARY SCHOOL STUDENTS
(GREEN GENIE)

GOH MEI KEI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy/Master of Science/Master of Engineering

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

ACKNOWLEDGEMENTS

I would like to express my greatest appreciation to the all individuals who have helped and supported me throughout the project. I am thankful to my supervisor, Dr. Danakorn Nincarean for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project. I would also like to thank my Computer Game Programming Professors, Dr. Saradatul Akmar and Dr. Tuty Asmawaty for teaching me game programming skills that I could apply to this game project.

A special acknowledgement goes to my friends, Leong Pi Yee, Lee Zi Xuan and Yong Rong See who helped me in completing the project by exchanging interesting ideas and sharing their experience. We spent countless nights working together on the same goal, and we worked side by side to make me feel that I was not alone in completing this project.

I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.

Once again, thanks to everyone for making this project successful.

ABSTRAK

Kelestarian alam sekitar adalah untuk menstabilkan hubungan tegang yang sedia ada antara tamadun manusia dan dunia kehidupan. Walau bagaimanapun, manusia sedang menyusut dan memusnahkan sumber bumi lebih cepat daripada yang boleh dipulihkan dan dijana semula. Sebab utama untuk ini adalah kerana kebanyakan individu mempunyai tahap kesedaran alam sekitar yang lemah mengenai kitar semula. Adalah penting untuk mula mengajar kanak-kanak tentang keperluan kitar semula dan menjalani gaya hidup mesra alam seawal mungkin supaya mereka mengetahui bahawa tidak semua atau semua sumber adalah tidak terhingga. Oleh itu, projek ini adalah untuk menjadikan idea itu sebagai permainan hiburan pendidikan untuk kanak-kanak berumur 7 tahun ke atas bagi mencetuskan minat mereka terhadap kitar semula dan keseimbangan alam sekitar dalam projek permainan ini. Objektif projek permainan ini adalah untuk mengkaji permainan kelestarian alam sekitar sedia ada untuk pelajar sekolah rendah, membangunkan permainan kelestarian alam sekitar yang menyediakan pengetahuan kitar semula untuk pelajar sekolah rendah dan menilai kefungsi permainan kelestarian alam sekitar yang dibangunkan untuk pelajar sekolah rendah. Permainan ini adalah permainan berasaskan komputer peribadi (PC) dan ia dibangunkan dengan menggunakan enjin permainan Unity. Genre permainan First-Person Shooter (FPS) akan dilaksanakan dalam projek permainan ini untuk meningkatkan minat kanak-kanak bermain permainan kitar semula. Semua item yang dikumpul oleh pemain akan ditukar menjadi peluru untuk Blasters dan menembak ke tong kitar semula yang sepadan. Hasil daripada projek ini ialah pemain boleh belajar tentang pengkategorian sampah dan berlatih melakukan kitar semula. Terdapat juga mesej tersembunyi yang disampaikan kepada pemain yang memaklumkan mereka bahawa pengurusan sisa yang tidak betul boleh membawa kepada penyebaran kuman dan membahayakan hidupan, terutamanya hidupan akuatik.

ABSTRACT

Environmental sustainability is to stabilize the existing strained relationship between human civilization and the living world. However, humans are depleting and destroying the earth's resources faster than they can be restored and regenerated. The major reason for this is because most individuals have a poor level of environmental awareness regarding recycling. It is critical to start teaching children about the need of recycling and living an eco-friendly lifestyle as early as possible so that they learn that not everything or all resources are infinite. Therefore, this project is to turn the idea into an educational entertainment game for children aged 7 and up in order to spark their interest in recycling and environmental concerns in this game project. The objectives of this game project is to study the existing environmental sustainability game for elementary school students, develop an environmental sustainability game that provides recycling knowledge for elementary school students and evaluate the functionality of the developed environmental sustainability game for elementary school students. The game is a personal computer (PC) based game and it develops by using Unity game engine. First-Person Shooter (FPS) game genre will be implemented in this game project to enhance the children's interest in playing the recycle game. All the items that collected by the players will convert into ammo for the Blasters and shoot to the corresponding recycle bin. The result through this project is that player can learn about the categorizing of garbage and practice to do the recycle. There are also hidden messages delivered to the players alerting them that improper waste management might lead to the spread of germs and harm living things, particularly aquatic life.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope	3
1.5 Thesis Organization	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Review of Existing Game Application	5
2.1.1 Lumino City	5
2.1.2 Plasticity	8
2.1.3 GreenSpace	10
2.2 Comparison Table of Three Existing Game Applications	11
2.3 Advantages and Disadvantages of Existing Game Applications	12
2.4 Chapter Summary	16

CHAPTER 3	METHODOLOGY	17
3.1	Introduction	17
3.2	Project Management Framework - RAD Model	18
3.3	Project Requirements	19
3.3.1	Terms and Acronyms	19
3.3.2	Functional Requirements	20
3.3.3	Non-functional Requirements	20
3.3.4	Constraints and Limitations	21
3.4	Proposed Design	22
3.4.1	Use Case Diagram	22
3.4.2	Context Diagram	23
3.5	Game Design Document (GDD)	23
3.5.1	Game Genre	24
3.5.2	Game World	24
3.5.3	Storyline	24
3.5.4	Level Design	25
3.5.5	Game Mechanics	25
3.5.6	Storyboard	28
3.5.7	Flowchart	38
3.5.8	Game Technology	39
3.6	Data Design	41
3.7	Testing Plan	42
3.7.1	Alpha Test	43
3.7.2	Beta Test	45
3.8	Potential Use of Proposed Solution	46

CHAPTER 4 RESULTS AND DISCUSSION	47
4.1 Introduction	47
4.2 Project Setup	47
4.3 Implementation Process	48
4.3.1 Designing Graphic and Multimedia Contents	49
4.3.2 Game Characters Control	68
4.3.3 Development of Game Application	81
4.3.4 Game Mechanic Setup	123
4.4 Results and Discussion	136
4.5 Chapter Summary	143
CHAPTER 5 CONCLUSION	144
5.1 Introduction	144
5.2 Limitation and Constraint	145
5.3 Future Work	146
REFERENCES	147
APPENDIX A	149
APPENDIX B	161

LIST OF TABLES

Table 2.1	Comparison of Three Existing Systems	11
Table 3.1	Functional Requirements	20
Table 3.2	Non-Functional Requirements	20
Table 3.3	List of Software Used in Green Genie	40
Table 3.4	List of Hardware Used in Green Genie	40
Table 3.5	Test Case of Main Menu	43
Table 3.6	Test Case of How To Play Menu	43
Table 3.7	Test Case of the Gameplay	44
Table 3.8	Test Case of Pause Menu	45
Table 4.1	Software to setup during the development of GreenGenie	47
Table 4.2	Essential Unity Packages Installed	48
Table 4.3	Images that are imported into Unity or Blender as a plane	49
Table 4.4	The details of 3D models included in the game project.	50
Table 4.5	The details and designing method for creating terrains.	53
Table 4.6	Before and after editing the original 3D models	68
Table 4.7	Screenshots of the player character's animation states	71
Table 4.6	Game Rules Interfaces for Level 1	91
Table 4.7	Game Rules Interfaces for Shooting the Germ	97
Table 4.8	Game Rules Interfaces for Level 2	101
Table 4.9	Game Rules Interfaces for Level 3	107

LIST OF FIGURES

Figure 2.1	Screenshot of Lumino City	5
Figure 2.2	Pre-production for creating the handmade paper world - Lumino City	6
Figure 2.4	One of the puzzle games used for developing films.	6
Figure 2.5	Screenshot of Gears mechanisms that unlocked the next site in the gameplay	7
Figure 2.6	900-page illustrated Handy Manual Book	7
Figure 2.7	Movement Control of Player in Plasticity	8
Figure 2.8	Screenshot of Plasticity	8
Figure 2.9	Screenshot of Plasticity	9
Figure 2.10	Screenshot of Plasticity	9
Figure 2.11	Screenshot of Plasticity	9
Figure 2.12	Gameplay of GreenSpace.	10
Figure 2.13	Gameplay of GreenSpace.	11
Figure 2.14	Positive Review on Steam	13
Figure 2.15	Positive Review on Steam	13
Figure 2.16	Negative Review on Google Play Store	14
Figure 2.17	Negative Review on Google Play Store	14
Figure 2.18	Negative Review on Steam	14
Figure 2.19	Positive Review on Steam	15
Figure 2.20	Positive Review on Steam	15
Figure 2.21	Negative Review on Steam	15
Figure 3.1	Rapid Application Development Life Cycle	18
Figure 3.2	Proposed framework based on RAD model	19
Figure 3.3	Use Case Diagram of Green Genie	22
Figure 3.4	Context Diagram of Green Genie	23
Figure 3.5	Overall Flowchart of GreenGenie	38
Figure 3.6	Data Design for developing GreenGenie	41
Figure 4.1	Modifier properties of the recycle bin models.	51
Figure 4.2	Fence model	51
Figure 4.3	Planting area	52
Figure 4.3	Waterfall shape model	52
Figure 4.4	Assigning waterfall material into the particle system under the render section.	52

Figure 4.5	Overview of Genie Land Terrain	53
Figure 4.6	Spirit Tree of Green Genies	53
Figure 4.6	Paint Texture Setting in Terrain Editor.	54
Figure 4.7	Paint Trees Setting in Terrain Editor.	55
Figure 4.8	The area that wants to paint tree	55
Figure 4.9	Results of Terrain after adding all the materials.	56
Figure 4.10	Results of Terrain after adding all the materials.	56
Figure 4.11	Overview of Trash Island Terrain	57
Figure 4.12	Paint Texture Setting in Terrain Editor.	57
Figure 4.13	Results of Terrain after adding all the materials.	58
Figure 4.14	Overview of Trash Fall Terrain	59
Figure 4.15	Paint Texture Setting in Terrain Editor.	59
Figure 4.16	Overview of Dead Tree Island Terrain	60
Figure 4.17	Paint Texture Setting in Terrain Editor.	61
Figure 4.18	Paint Grass Setting in Terrain Editor.	62
Figure 4.19	Results of Terrain after adding all the materials.	62
Figure 4.20	Waterfall for Scene 3 Game Environment	63
Figure 4.21	Property of Waterfall Shader Graph	63
Figure 4.22	Screenshots of Shader Graph built for Waterfall Splashing Effect	64
Figure 4.23	Result of WaterfallShader	64
Figure 4.24	Property of ScrollShader Graph	65
Figure 4.25	Ripple Lines	65
Figure 4.26	Screenshots of Shader Graph built for Scroll Shader Effect for ripples	66
Figure 4.27	Result of ScrollShader	66
Figure 4.28	Audios in the game	66
Figure 4.29	Standard text in the game	67
Figure 4.30	Text mesh pro in the game	67
Figure 4.31	Screenshot of the Animator of the main character in Unity.	71
Figure 4.32	Basic Animation - idle, walk, run, attack	78
Figure 4.33	Texture Paint in Blender	79
Figure 4.34	Armature bone is added to the Sea turtle 3D model	79
Figure 4.35	Basic Animation - idle, walk, run, eat	79
Figure 4.31	Main Menu User Interface	81
Figure 4.32	First panel in How To Play Interface	82

Figure 4.33	Second panel in How To Play Interface	82
Figure 4.34	Second panel in How To Play Interface	82
Figure 4.35	Second panel in How To Play Interface	83
Figure 4.36	Second panel in How To Play Interface	83
Figure 4.37	Second panel in How To Play Interface	83
Figure 4.38	Screenshot of the storyline of the Introduction scene.	84
Figure 4.39	Screenshot of the storyline of the Introduction scene.	85
Figure 4.40	Screenshot of the storyline of the Introduction scene.	85
Figure 4.41	Dialogue between Jinn and small genie	85
Figure 4.42	Dialogue between Jinn and small genie	86
Figure 4.43	Dialogue between Jinn and small genie	86
Figure 4.44	Dialogue between Jinn and small genie	86
Figure 4.45	Dialogue between Jinn and small genie	87
Figure 4.46	Dialogue between Jinn and small genie	87
Figure 4.47	Dialogue between Jinn and small genie	87
Figure 4.48	Dialogue between Jinn and small genie	88
Figure 4.49	Animation for the First Scene Camera	88
Figure 4.50	First View of the player character in Tutorial Scene.	89
Figure 4.51	Tutorial to guide the player to switch weapons.	89
Figure 4.52	Tutorial to guide the player to collect bottles.	90
Figure 4.53	Tutorial to guide the player to shoot bottles to the recycle bin.	90
Figure 4.54	Tutorial to guide the player to kill germs.	90
Figure 4.55	Pause Menu User Interface	91
Figure 4.56	View of Trash Island with Orange Bin and sea turtles from the opposite Island.	93
Figure 4.56	Wooden board with tips in Level 1	93
Figure 4.57	Guidelines or tips interfaces for the player.	94
Figure 4.58	Sea turtles swim up to help the player jump across the sea.	94
Figure 4.58	Drowning Interface	94
Figure 4.59	Drowning Animation	94
Figure 4.60	Script that assigned to the Character in the Inspector.	95
Figure 4.61	Creating the path with Path Creator Script	95
Figure 4.63	The path for the Green light ball to follow.	96
Figure 4.64	The end result of the Green light ball follows the path.	96
Figure 4.65	View of Trash Island with Germs from the opposite Island.	97

Figure 4.66	Objects drop from the Germs after getting killed by the player.	98
Figure 4.67	Mission that need to be completed in Level 1.	98
Figure 4.68	The message interface to notify the player teleportal is accessible.	99
Figure 4.69	Game Over Canvas shown when the turtle is dead.	100
Figure 4.70	Game Over Canvas shown when times up or the player killed by the Germs.	100
Figure 4.71	Victory Canvas when player complete all the mission and get into the teleportal.	100
Figure 4.72	Game View of Level 2	102
Figure 4.74	Tree planted by the player in Level 2.	102
Figure 4.75	Game View of Level 2	103
Figure 4.76	To cross the river, the player must leap onto the large stone.	103
Figure 4.77	Drowning Interface	103
Figure 4.78	Drowning Animation	104
Figure 4.79	Wooden board with tips in Level 2	104
Figure 4.80	Mission that needs to be completed in Level 2.	104
Figure 4.81	The message interface to notify the player teleportal is accessible.	105
Figure 4.82	Game Over Canvas shown when times up or the player killed by the Germs.	106
Figure 4.83	Victory Canvas when player complete all the mission and get into the teleportal.	106
Figure 4.84	Game View of Level 3	108
Figure 4.85	Wooden board with tips in Level 3	108
Figure 4.86	Wooden board with tips in Level 3	108
Figure 4.87	Tree planted by the player in Level 3.	109
Figure 4.88	Mission that needs to be completed in Level 3.	109
Figure 4.89	The message interface to notify the player teleportal is accessible.	110
Figure 4.90	Game Over Canvas shown when times up or the player killed by the Germs.	111
Figure 4.91	Victory Canvas when player complete all the mission and get into the teleportal.	111
Figure 4.92	Game Rules Interfaces for Level 4	112
Figure 4.93	Game View of Level 4	112
Figure 4.94	Tips messages of recycling plastic bottles in Level 4	113
Figure 4.95	Tips messages of recycling boxes/cardboard in Level 4	113
Figure 4.96	Tips messages of recycling glass bottles in Level 4	113
Figure 4.97	Mission that needs to be completed in Level 4.	114

Figure 4.98	Collectable health drop from the Germs after getting killed.	115
Figure 4.99	The message interface to notify the player teleportal is accessible.	115
Figure 4.100	Game Over Canvas shown when times up or the player killed by the Germs.	116
Figure 4.101	Victory Canvas whe player complete all the mission and get into the teleportal.	116
Figure 4.102	Game Rules Interfaces for Level 5	117
Figure 4.103	Game View of Level 5	117
Figure 4.104	Mission that needs to be completed in Level 5.	118
Figure 4.105	Collectable health drop from the Germs after getting killed by the player.	119
Figure 4.106	The message interface to notify the player teleportal is accessible.	119
Figure 4.107	Game Over Canvas shown when times up or the player killed by the Germs.	120
Figure 4.108	Victory Canvas whe player complete all the mission and get into the teleportal.	120
Figure 4.109	End Scene Interface	121
Figure 4.110	End Scene Interface	121
Figure 4.111	End Scene Interface	121
Figure 4.112	End Scene Interface	122
Figure 4.113	End Scene Interface	122
Figure 4.114	Script Components in a GameObject named “Character”	123
Figure 4.115	Controller component assigned to Character GameObject	124
Figure 4.116	Script for Player Movement Control	124
Figure 4.117	Script for picking up weapon	125
Figure 4.118	Script for changing weapon	125
Figure 4.119	Script for collect loot drop objects	125
Figure 4.120	Script for change ammo/ collected objects	125
Figure 4.121	Script for locking the cursor	126
Figure 4.122	Script for displaying the cursor while the pause menu shows up	126
Figure 4.123	Script for checking whether NPC can see the player in State.cs	126
Figure 4.124	Script for checking whether NPC is close to the player in State.cs	127
Figure 4.125	Script for being idle in State.cs	127
Figure 4.126	Script for patrolling in State.cs	127
Figure 4.127	Script for pursuing target in State.cs	128
Figure 4.128	Script for attacking player in State.cs	128
Figure 4.129	Script for picking items.	129

Figure 4.130	Script for filling the progress bar with gradient colour.	129
Figure 4.131	Hierarchy of Germ gameObject	130
Figure 4.132	Adding Capsule Collider to Germ's parent child - Collide gameObject	130
Figure 4.133	Script for health bar	130
Figure 4.132	Script for the ammo hitting the collider of bin to add points	131
Figure 4.133	The points added can be assigned at point value	131
Figure 4.134	The scripts for adding score if the ammo hit the correct colour of recycle bin	131
Figure 4.135	The scripts for deducting score if the animal is die	131
Figure 4.138	The scripts for getting result display	132
Figure 4.139	The GameDatabase scriptable object	133
Figure 4.140	The ammo database scripts for initialize the variables	133
Figure 4.141	The ammo database scripts for getting entries	133
Figure 4.142	Set the Ammo GameObject as public.	134
Figure 4.143	Script for object loot drop	134
Figure 4.144	Script assigned to Germ GameObject	134
Figure 4.146	Scripts for Pause Menu	135
Figure 4.147	Scripts for Controlling load scene and main menu function.	135
Figure 4.148	Scripts for Saving Checkpoints	136
Figure 4.149	A 11-year-old girl named Teh Min Min playing Green Genie game	137
Figure 4.150	A 12-year-old girl named Lee Tze Qian playing Green Genie game	137
Figure 4.151	A 5-year-old girl named Yeong Min Fei playing Green Genie game	137
Figure 4.152	A 13-year-old boy named Yeong Zi Qing playing Green Genie game	138
Figure 4.153	A 12-year-old girl named Yeong Pei Ying playing Green Genie game	138
Figure 4.153	Result of "What is your highest education level?"	139
Figure 4.154	Result of "Does respondent think Green Genie's gameplay is fun and interesting?"	139
Figure 4.155	Result of "The gameplay of Green Genie is easy to understand? "	140
Figure 4.156	Result of "Green Genie is suitable for elementary students to play?"	140
Figure 4.157	Result of "Green Genie has enhanced recycling knowledge?"	141
Figure 4.158	Result of "Green Genie has raised the environmental awareness?"	141
Figure 4.159	Result of "Which Level of Green Genie do you think is the most difficult to pass?"	142
Figure 4.160	Result of "Kindly leave a short comment(s) on this game."	143

CHAPTER 1

INTRODUCTION

1.1 Introduction

According to a well-known American environmentalist, Paul Hawken, the goal of environmental sustainability is to stabilize the existing strained relationship between the earth's two most complex systems: human civilization and the living world (Paul Hawken, 2008). To be more precise, a sustainable ecosystem is a pace at which renewable resources are extracted and non-renewable resources depletion can be sustained eternally. Paul Hawken has also written in his book, *Blessed Unrest*, about the realization that humans are depleting and destroying the earth's resources faster than they can be restored and regenerated [1]. The main reason behind this fact is the low level of environmental awareness on recycling among most people. It has been challenging to effectively control recycling habits, especially among households.

One of the key factors that cause waste accumulation in some countries is the growing population and pursuit of contemporary lifestyles. For instance, Malaysia is one of the countries that face the issue of the rapid growth of domestic waste collected from households (Afroz, Masud, Akhtar & Duasa, 2013). Based on the data they collected from the study of proper management of waste from electrical and electronic equipment (WEEE), only 3% and 2% of the households were involved in the recycling campaign of their WEEE. Even though the Malaysian government has launched many policy evolution of solid waste management, those plans and policies were not implemented effectively. This is because most of the citizens lack knowledge in handling the separation of household wastes.

Everyone on the world must be environmentally conscious since everyone bears responsibility for maintaining the environment and promoting the planet's long-term

growth for future generations. As a result, it is critical to start teaching children about the need of recycling and living an eco-friendly lifestyle as early as possible so that they learn that not everything or all resources are infinite. This idea will be turned into an educational entertainment game for children aged 7 and up in order to spark their interest in recycling and environmental concerns in this game project. This project intends to produce an entertaining and instructive game that may re-energize students' desire for learning and explore the joy and thrill of learning.

1.2 Problem Statement

Environmental issues such as overflowing landfills, pollution, environmental degradation and resource depletion are a matter of concern nowadays. Landfills are the designated place to deposit waste. Due to the rapid growth of the population and economy, most people continue to neglect to recycle. Littering has become the factor that causes our landfills to continuously pile up, emitting terrible smells and depositing hazardous chemicals into the land and sky.

In addition, the separation of household wastes is uncommon in developing countries including Malaysia. This is due to a low level of environmental awareness and negative perception of waste handling (Innocent A. Jereme, I. A., Siwar, C. & Alam, M. M. (2015). To support this statement, an article published in *The New Sunday Times* described the Malaysian attitude toward recycling as a Not-In-My-Backyard (NIMBY) syndrome when it comes to waste management issues [4]. Most of them choose not to adopt recycling activities because they feel troublesome and also do not have much recycling knowledge.

Furthermore, while most governments, including Malaysia's, have committed to achieving green growth and outlined ongoing efforts to establish a resource-efficient and sustainable economy, challenges remain in improving environmental quality and conserving natural resources, particularly in terms of unsustainable development and low environmental awareness. One reason for this might be because the government's recycling initiatives or propaganda efforts are not appealing enough. The government should incorporate the entire issue into the school system for young children, because healthy habits must be taught and developed at a young age.

As a result, the purpose of this project is to develop an educational entertainment game that is related to environmental sustainability. Recycling is one of the effective ways to help to stop our natural resources from running out. Therefore, this game project will be offering the recycling idea and knowledge on how to manage the separation of waste. This game project is also planned to be designed for children aged between 7 to 12 years old. This is because good habits such as recycling and knowledge about the importance of environmental sustainability should be developed from an early age.

1.3 Objectives

There are three objectives in this project which are:

- 1) To study the existing environmental sustainability game for elementary school students.
- 2) To develop an environmental sustainability game that provides recycling knowledge for elementary school students.
- 3) To evaluate the functionality of the developed environmental sustainability game for elementary school students.

1.4 Scope

The scopes in this project are:

- 1.) This project will be created utilizing Unity.
- 2.) The project will feature 6 types of multimedia information to boost its appeal: video, audio, 3D models, images, and annotations.
- 3.) This project will be developed in a PC game format.
- 4.) This project will include 5 levels that uses educational materials about environmental and recycling knowledge to design each of the levels.
- 5.) This project will be providing tips to players on waste management.
- 6.) The target user for this project is planned to be designed for children aged between 5 to 12 years old.

1.5 Thesis Organization

This thesis is divided into five chapters. Chapter 1 discussed the introduction of this project. This chapter described the game's introduction and how it was infused with environmental sustainability, problem statements, objectives, and scopes of the project to be created.

Chapter 2 discussed the literature review. This chapter described the research done on three different existing games that are related to this project idea. The game features such as the gameplay method, platform, techniques, hardware and software used in these games are also studied in this chapter.

Chapter 3 analyzed the methodology that will be used in this project. The software development model, game design, game technology and technique used, approach and flow of this project will be discussed and explained in this chapter.

Chapter 4 is about the result and discussion of this project. The process of game development and implementation such as game interfaces and game functionality will be explained in detail together with the testing results and discussion.

Lastly, the conclusion and summary of this project will be concluded in Chapter 5 which also includes the limitation and future work that can be implemented to improve this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Review of Existing Game Application

There are three existing systems on the market related to this project that will be evaluated and compared in this chapter. The game features, the core gameplay, techniques, strengths and weaknesses of the studied systems will be discussed discreetly in this section. The games selected are all environmental games that are suitable for children to play. These environmental games are able to transform traditional education into an effective learning experience for both students and teachers.

2.1.1 Lumino City

Lumino City is a single-player handcrafted puzzle adventure game made completely of paper, cards, microlights, and motors. The storyline of the game is the main game character, Lumi, who found out her Grandpa had been kidnapped. Her Grandpa was the caretaker of Lumino City. To find him, players have to explore the city and solve all the puzzles to figure out the fascinating mechanisms that power this Lumino World.



Figure 2.1 Screenshot of Lumino City



Figure 2.2 Pre-production for creating the handmade paper world - Lumino City

Lumino City is crafted by State of Play and Noodlecake Studios and has won numerous International Awards including BAFTA for Artistic Achievement, Develop Award, Indie Prize Casual Connect, British Academy Games Award and more. The players are playing with real objects within each of the scenes in Lumino City. The gameplay involves a variety of inspiring and challenging puzzles such as providing power to the house using only lemon and ingenuity of players and a lot more to explore.

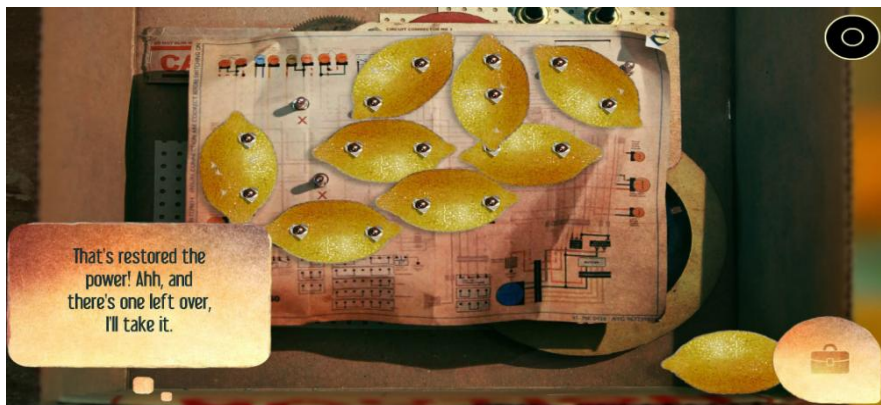


Figure 2.3 First puzzle game related to renewable technology.



Figure 2.4 One of the puzzle games used for developing films.

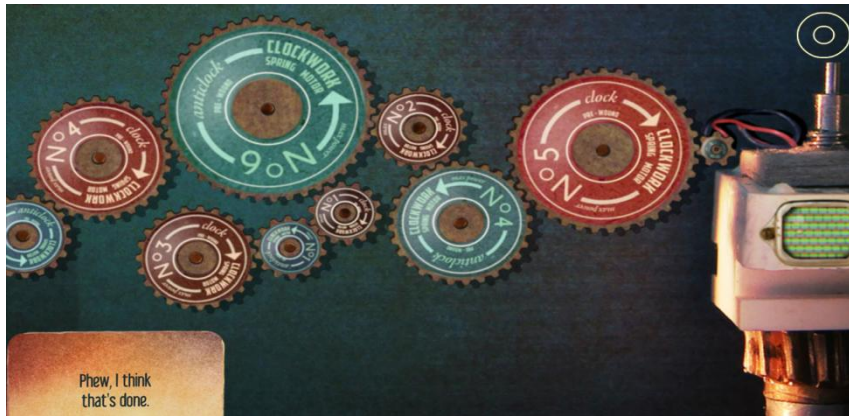


Figure 2.5 Screenshot of Gears mechanisms that unlocked the next site in the gameplay

There are many of the mini puzzle tasks designed in Lumino City that involve using renewable technology to power a site or exploring a type of self-sufficiency. After the player solves a task, the following site will be unlocked. The player has to go through all the sites in Lumino City to reach the goal which is finding the game character, Lumi's Grandpa.



Figure 2.6 900-page illustrated Handy Manual Book

The Handy Manual Book will act as the hint given to the players throughout the game and explore a variety of applications of renewable energy technologies. It has 900 illustrated pages but only 40 of which are necessary for the game. Therefore, the players have to flip through to seek solutions in every puzzle and problem.

From this game, players are able to raise awareness about renewable energy technology and learn to incorporate sustainable behaviour into the gameplay.

2.1.2 Plasticity

Plasticity is a puzzle-platformer video game about a plastic-ridden world. The main character of the game is Noa who flees her dreary home in pursuit of paradise and a brighter future.



Figure 2.7 Movement Control of Player in Plasticity



Figure 2.8 Screenshot of Plasticity



Figure 2.9 Screenshot of Plasticity



Figure 2.10 Screenshot of Plasticity



Figure 2.11 Screenshot of Plasticity

Based on Figures 2.8 to 2.11, the players begin on a journey with flooded cities and ravaged lands. Players need to make decisions to save the world. Therefore, every decision that is made will carry consequences and impact both the gameplay and the plot of the storyline.

Throughout the game, players will think about the trash they throw away and the plastic they recycle. Plasticity will then reveal how the action makes an impact through the art of storytelling. The message that the game wants to deliver is the importance of waste management and recycling.

2.1.3 GreenSpace

GreenSpace is an environmentally-conscious game that can be played through the Facebook social network and is free with in-app purchases. The storyline of the game is about humans who have sent trash into the universe for the past 300 years and caused serious pollution in outer space. Players will act as GreenSpace Corps and be sent into the galaxy to clean up the space and make them habitable again. Once the players clean up the trash in a quadrant of the island, they will acquire Rocket Fuel, Credits, Energy, and experience points. These elements help the players to unlock new quadrants, build new and repair structures, and complete the missions.



Figure 2.12 Gameplay of GreenSpace.



Figure 2.13 Gameplay of GreenSpace.

GreenSpace raises environmental awareness to the players and encourages them to know the importance of sustainability through waste management gameplay. After developing this game, there are very interesting and meaningful deeds done by the developer team, RocketOwl. They had partnerships with the reforestation organization, WeForest.org. WeForest will sprout trees to reflect players' achievements in GreenSpace when they complete in-game milestones.

2.2 Comparison Table of Three Existing Game Applications

Based on the review done earlier, Table 2.1 shows the comparisons of three existing game applications. Eleven aspects such as game platform, target system, operating system (OS), game engine, game genre, download cost, details of in-app purchase, graphics, network access, player mode and difficulty of the game applications will be reviewed in the table.

Table 2.1 Comparison of Three Existing Systems

	Lumino City	Plasticity	GreenSpace
Platform	<ul style="list-style-type: none"> ● Steam ● PC ● Mobile 	<ul style="list-style-type: none"> ● PC 	<ul style="list-style-type: none"> ● Facebook social network platform
Target System	<ul style="list-style-type: none"> ● Microsoft Windows ● OSX ● Android ● iOS 	<ul style="list-style-type: none"> ● Microsoft Windows 	<ul style="list-style-type: none"> ● Microsoft Windows ● Android ● iOS

Operating System	<ul style="list-style-type: none"> ● Windows ● Processor: Intel 2GHz ● Memory: 2 GB RAM ● OSX ● Processor: Intel 2GHz ● Memory: 2 GB RAM ● Android 4.0 and above ● iOS 10.0 or later 	<ul style="list-style-type: none"> ● Windows 7 and above ● Processor: IntelCore 2 Quad Q6600 @ 2.4 GHz, AMD FX 8120 @ 3.1 GHz ● Memory: 4 GB RAM 	<ul style="list-style-type: none"> ● Windows 7 or higher ● Processor: Duo Core or higher ● Memory: 1GB ● Android 2.2 and above ● iOS 4.3 or later
Game Engine/ Techniques to develop	<ul style="list-style-type: none"> ● Made entirely by craft materials such as paper, card, wood, miniature lights and motors ● State-of-play in-house software 	<ul style="list-style-type: none"> ● Unity ● FMOD (Sound effect engine) ● MAYA for character modelling 	<ul style="list-style-type: none"> ● Adobe Flash
Game Genre	<ul style="list-style-type: none"> ● Puzzle-platformer ● Casual ● Indie ● Adventure 	<ul style="list-style-type: none"> ● Puzzle-platformer ● Roleplay ● Educational 	<ul style="list-style-type: none"> ● Simulation ● Educational
Download Cost	<ul style="list-style-type: none"> ● RM30 (Steam) ● RM 23.99 (Google PlayStore & App Store) 	<ul style="list-style-type: none"> ● Free 	<ul style="list-style-type: none"> ● Free
In-app purchase	<ul style="list-style-type: none"> ● No 	<ul style="list-style-type: none"> ● No 	<ul style="list-style-type: none"> ● Yes
Graphics	<ul style="list-style-type: none"> ● Handcrafted 2D 	<ul style="list-style-type: none"> ● 3D 	<ul style="list-style-type: none"> ● 3D
Network Access	<ul style="list-style-type: none"> ● No 	<ul style="list-style-type: none"> ● No 	<ul style="list-style-type: none"> ● Yes
Player Mode	<ul style="list-style-type: none"> ● Single-player 	<ul style="list-style-type: none"> ● Single-player 	<ul style="list-style-type: none"> ● Single-player ● Multiplayer interaction
Difficulty	<ul style="list-style-type: none"> ● Medium-Hard 	<ul style="list-style-type: none"> ● Easy-Medium 	<ul style="list-style-type: none"> ● Easy

2.3 Advantages and Disadvantages of Existing Game Applications

Lumino City has a very impressive and cool art style since the architecture of the whole game is handcrafted. The gameplay of Lumino City is challenging yet

healing since it is a combination of animation with many mini-puzzle problems. In addition, there is a Handy Manual book that recorded a lot of knowledge about renewable energy technologies which can raise the environmental awareness of the player. Besides, the soundtrack of the game is very comfortable and relaxed. Therefore, it might be the main reason that the player still can play the game with peace of mind even though they might spend a long time solving the puzzles.

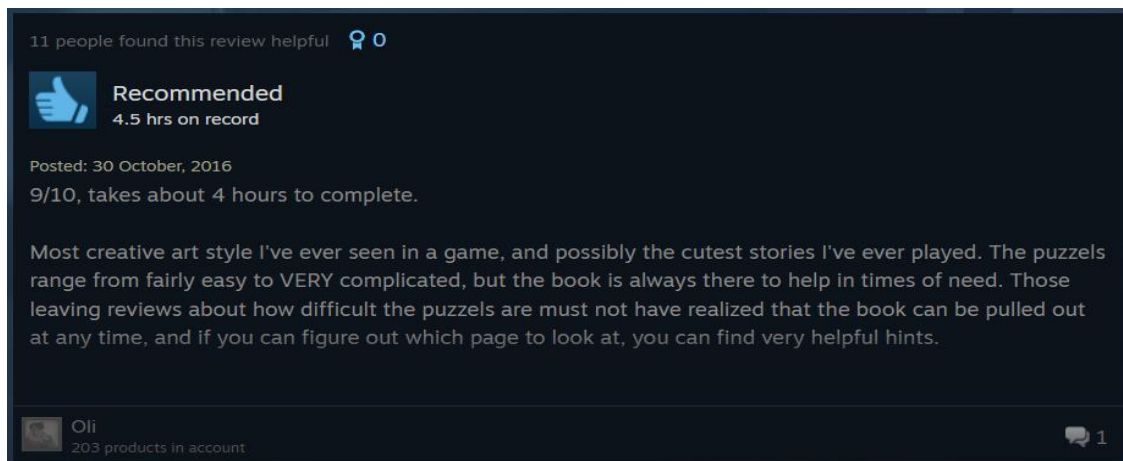


Figure 2.14 Positive Review on Steam



Figure 2.15 Positive Review on Steam

However, Lumino City also has its weak points. Small parts of the players had complained about the issues of character movement. Since most of the time, they need more time to explore where to go, and even worse, when they click the wrong spot, they have to reach the wrong spot only so they can begin to choose a new spot. This might

cause those impetuous players to lose patience and feel time-consuming while playing the game. Besides, it has issues with the resolution of the game being glitchy and buggy for some devices starting from February 2022.

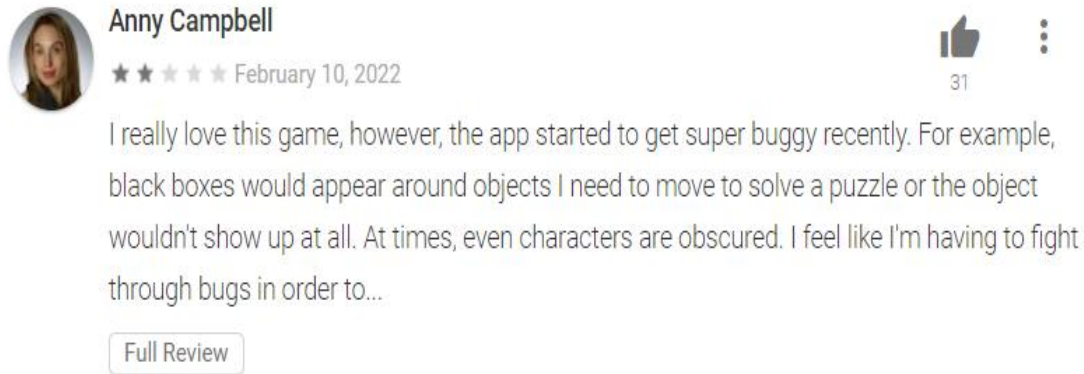


Figure 2.16 Negative Review on Google Play Store

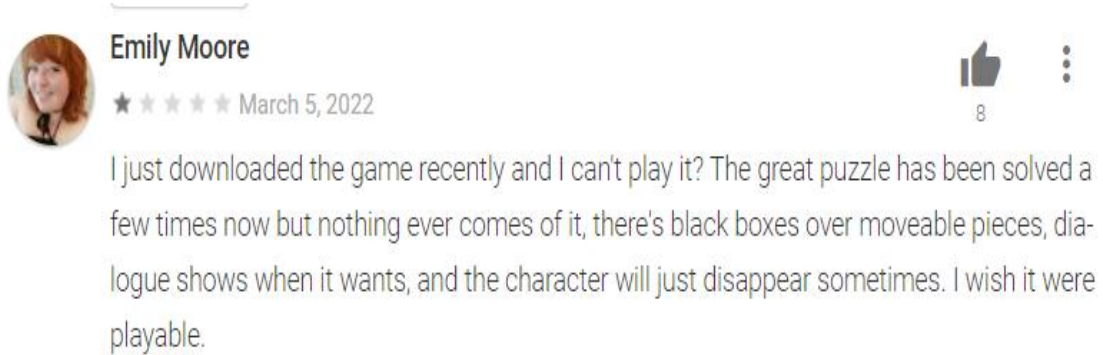


Figure 2.17 Negative Review on Google Play Store



Figure 2.18 Negative Review on Steam

Plasticity has nice graphics and a heart-touching storyline in the game. In addition, it is free to play. It also has eloquently expressed the damaging, hidden after-effects of plastics on the ecosystem, seas, and health repercussions, both graphically and in the tale being presented. Therefore, the message of protecting the environment will be successfully delivered to every player.

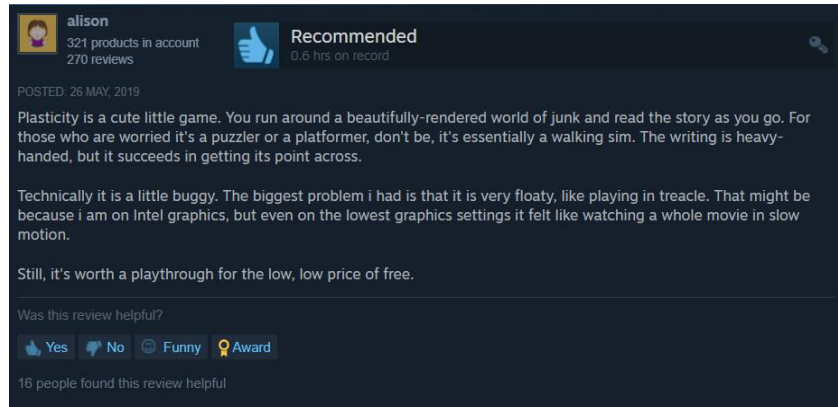


Figure 2.19 Positive Review on Steam



Figure 2.20 Positive Review on Steam

Nonetheless, plasticity has a few technical issues that need to be improved which are the clipping issues with the character and the world, and music will be glitchy in some sections, movement controls are stiff, buggy and slow control.

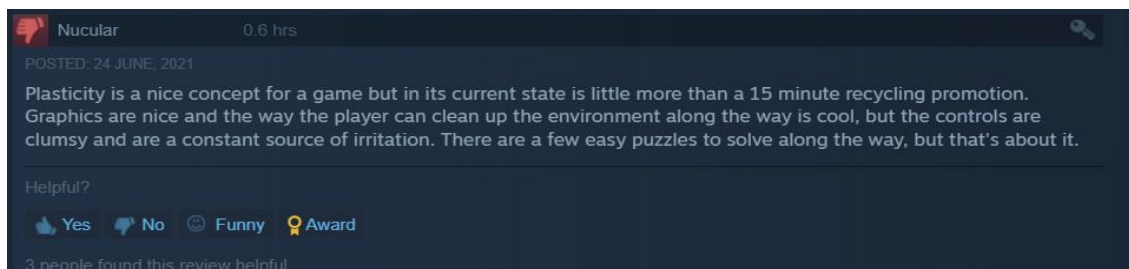


Figure 2.21 Negative Review on Steam

Furthermore, GreenSpace provides an engaging storyline that brings out a positive environmental message through waste management gameplay. It allows players to connect with their Facebook friends in-game as part of a cooperative effort to make a good influence on the environment. Besides being free to play, it has cute and friendly graphics, audio and gameplay.

The main problem with GreenSpace is it has stopped updating the latest version and was taken down recently even though it was a very nice and meaningful game.

2.4 Chapter Summary

Based on the analysis of the three existing game applications, an engaging storyline of the game application is important to attract more players. For instance, the main selling point and the most attractive part of Lumino City, Plasticity and GreenSpace are the engaging storylines and also the impressive graphics design in the game environment. Therefore, the designs for this project game application should not be overlooked and also implementation of a game application with an interesting storyline is a must to attract more players

Besides, suitable background music is also one of the crucial elements that have to be considered in implementing a game application. Based on the analysis, the majority of players who rated positive ratings commented that they enjoyed the game due to the graphics and music which make them feel delighted and comfortable. Therefore, choosing appropriate background music that suits the theme of the game application is also a point to note.

Last but not least, Plasticity got the most comments on the buggy issues from players compared to Lumino City and GreenSpace. However, those complaints in the reviews about the connection or difficulty in movement issues are not getting responses from the developer team. The simple ignoring might end up with tons of uninstalls and lowering the app ratings. Therefore, to develop a successful game application, getting feedback, fixing bugs, testing and making improvements consistently are essential things that cannot be ignored.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Software Development Life Cycle (SDLC) is a structured framework used to develop and manage applications systematically and cost-effectively. The process includes five phases which are planning and analysis, design, development, testing and implementation and also maintenance. The Waterfall model and Agile model are the popular SDLC methodologies that have been widely used by developers.

The Waterfall model is a linear and sequential life cycle that focuses on documentation and low customer involvement. Once the functional and non-functional requirements are defined in the early stage of the project lifecycle, the development process is strongly focused on meeting those agreed requirements which are risky for mistakes and not suitable to be used for projects where requirements are at a moderate to high risk of changing.

The Agile model is a much more flexible and adaptive methodology compared to the Waterfall model since it is incremental and includes iterative development in short sprints. It breaks the requirements into small incremental modules and combines those process models at the end with a focus on customer satisfaction by rapid delivery of the end product. However, it does not make sense when there is only one developer on the development project.

As a result, the appropriate SDLC that I choose to use in this project is the Rapid Application Development (RAD) model which is often used for small to medium projects that are time-sensitive.

3.2 Project Management Framework - RAD Model

Rapid Application Development (RAD) is a condensed development model that focuses on prototyping and enables the developer to quickly adapt to changing requirements and make multiple iterations without starting from scratch each time. This helps to make sure that the final product delivered is more quality-focused in the time constraint.

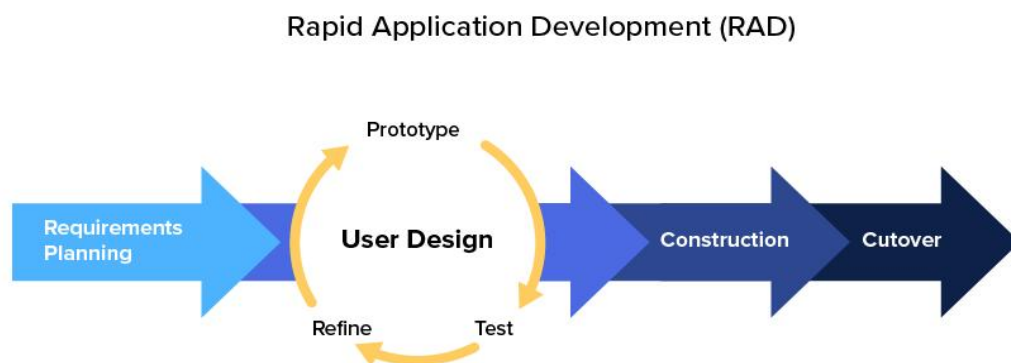


Figure 3.1 Rapid Application Development Life Cycle

There are four phases included in RAD which are defining project requirements, prototyping, rapid construction and implementation.

In the first phase, requirements such as project goals, expectations, timelines and potential issues have to be determined and identified. For example, the game purpose, and functional and non-functional requirements that are needed for this project will be determined before moving to the next stage. The detailed requirements will be further discussed in the next Project Requirement topic. (Refer to Table 3.1 and Table 3.2) In an addition, the potential issues such as the constraint and limitations of the game application will also be discussed in the following section. (Refer to Subtopic 3.3.4)

Once the project has been scoped, the initial model and prototype can begin to build out. The goal of the second phase is to produce a working design that can be demonstrated. At this stage, the storyboard and level designs of this game application along with the development of the prototypes will take place simultaneously until the project requirements are sufficiently met. After the prototype is done, reviews and feedback from the game tester or evaluators are needed to improve the prototype.

Besides, game testing to discover the bugs in the game development is important for this stage to improve the stability and performance of the game applications for this project. Therefore, errors will be discovered earlier which leads to a reduction in debugging and better structure for design addition in the future implementation stage.

Rapid construction is the third stage that includes coding, system testing and unit integration. If there are any changes or suggestion alterations discovered in this stage, the prototyping process of the game project still can be resumed until the feedback is strictly positive then only it will be moved on to the final step.

The final cut over phase is also known as the implementation phase. After completely addressing the technical defaults of this game project and optimizing implementation to enhance stability and maintainability, the game application for this project can be finalized and prepared for launch. Figure 3.2 shows the proposed framework for this game project based on the RAD model.

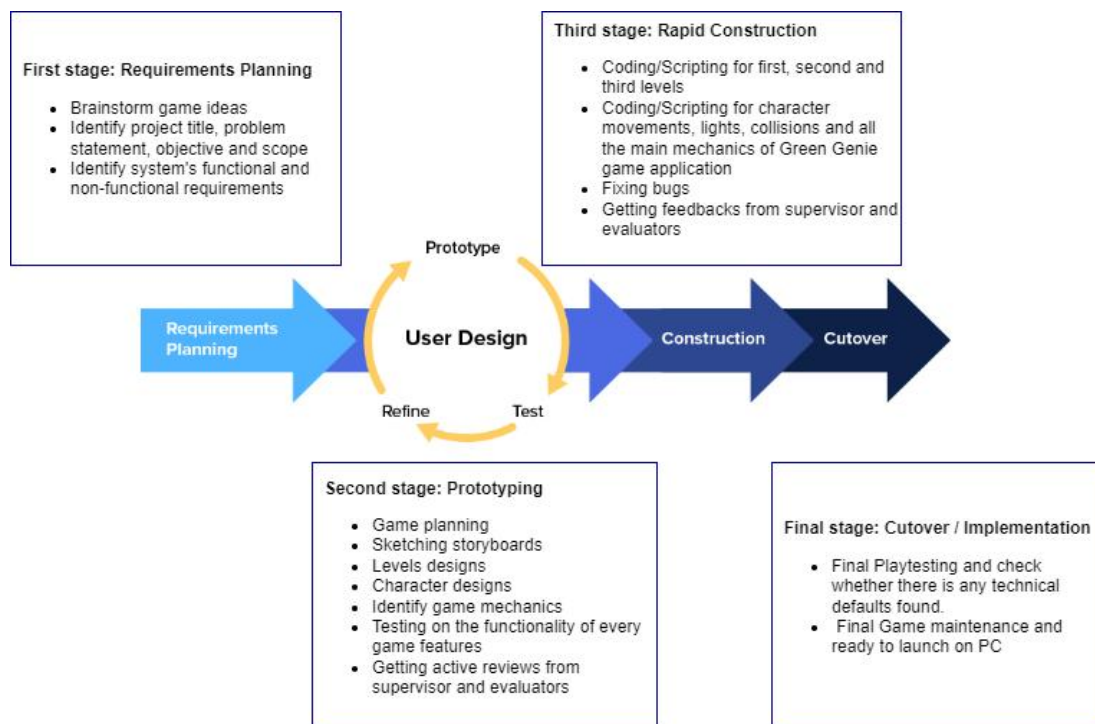


Figure 3.2 Proposed framework based on RAD model

3.3 Project Requirements

3.3.1 Terms and Acronyms

REQ = Requirements

GDD = Game Design Document

3.3.2 Functional Requirements

Functional requirement is a function of a system or its component, where a function is described as a specification of behaviour between inputs and outputs. Table 3.1 shows the functional requirements needed for this project.

Table 3.1 Functional Requirements

REQ-1:	The maximum time period from launching the game until it is playable should be 3 minutes. Players will have less than 3 minutes to load their data and resume playing the game from where they last left off. This is to ensure that the users will not lose their interest and patience in the waiting time.
REQ-2:	Prior to beginning the game, the system of application will evaluate fundamental computing abilities and offer the necessary tutorials for players.
REQ-3:	The system of application can be used fully with the output devices such as mouse and keypad for web/ screen touch for mobile devices.
REQ-4:	The game environment includes sewers, house buildings, cities, islands and rivers.
REQ-5:	The game theme should be fixed to environmental awareness.
REQ-6:	Game mechanics should include an inventory system, scoring system and countdown timer feature.

3.3.3 Non-functional Requirements

Non-Functional Requirement is the requirement that describes the operation of the system and the capabilities that improve its functionality. Table 3.2 shows the non-functional requirements needed for this project.

Table 3.2 Non-Functional Requirements

Non-Functional Requirement	Description
Usability	The system will provide in-game tutorials and help. The

	game needs to provide tutorials about recycling techniques for players so that players are able to learn and continue to play the game even if they lack environmental protection knowledge.
Reliability	The system of application will provide online backup. If the game crashes, the application will apply mechanics to guarantee that no data is lost.
Performance	The system of application should be able to perform smoothly and repeating materials that the player has mastered should not be required. For example, The users' progress should be automatically saved when the goals/level are completed. The users can run the application smoothly.
Security	Users are safe to install the game application. The game application should be free of viruses before and after being released to the market.

3.3.4 Constraints and Limitations

- **Device Compatibility**

Some user devices are not compatible to play the game as the game has a minimum requirement to execute the application. The minimum requirement for user devices is as follows:

- Windows 7 or newer: at least Dual-core from Intel or AMD at 2.8 GHz, minimum 4GB RAM, NVIDIA GeForce 8600/9600GT, ATI/AMD Radeon HD2600/3600 graphics card, Version 9.0c DirectX, DirectX Compatible Sound card and Broadband Internet connection is required.
- macOS X Mavericks 10.9 or newer: Dual-core from Intel, at least 4GB RAM or better, NVIDIA 320M or higher, or Radeon HD 2400 or higher, or Intel HD 3000 or higher, at least 15GB free space storage on the hard disk and Broadband Internet connection is required.

- **Technology Knowledge**

- All users will not have the same knowledge of using computers. It depends on the user and how often the user can access and use it. Differences in

employment or age group also can provide a large gap in terms of technological knowledge.

- Environmental Knowledge

- Users who have low environmental protection awareness might feel difficult to accomplish the tasks in the game application which affects the game experience.

3.4 Proposed Design

3.4.1 Use Case Diagram

A use case diagram is a graphical diagram that can be used to summarize the details of the users, also known as actors and their possible interactions with the system. There are several interactions in this project game application that can be done by users.

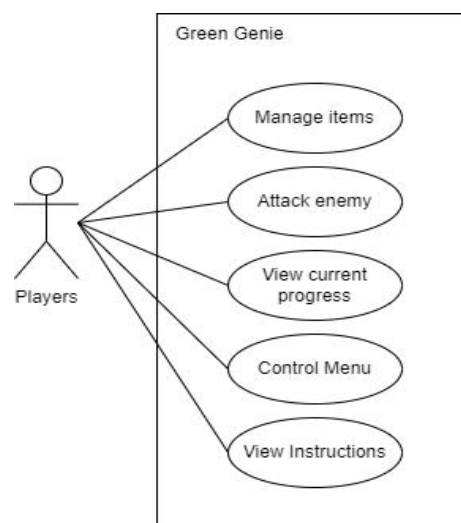


Figure 3.3 Use Case Diagram of Green Genie

Figure 3.3 shows the use case diagram of the Green Genie game application. The players are able to manage items such as collecting or storing items and also throwing the items into the correct recycle bin during the game. The players are also able to view the current recycling progress and control the game function such as adjusting volume, pausing and resuming the game through the game setting menu. Instructions or tutorial panels are also provided for players to view and refer to.

3.4.2 Context Diagram

A context diagram is a diagram that shows the scope and boundary between the system at a glance. It denotes the flow of data between the system and its external components.

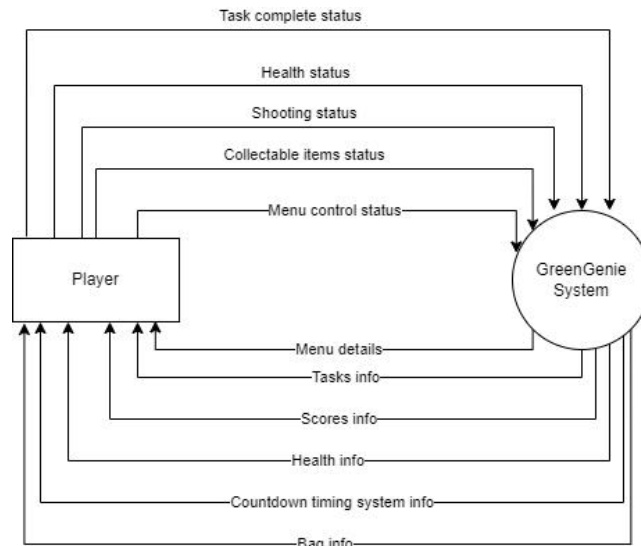


Figure 3.4 Context Diagram of Green Genie

Figure 3.4 shows the context diagram of Green Genie. Task information will be given by the system to players. For example, players have to accomplish the task by collecting all the items into the correct recycle bin in the limited time set. After the players complete the task, the status of the item such as collecting or storing items will be received by the system. Then the bag information will be updated for the players. Besides, item status such as throwing the items into the correct or incorrect recycle bin will be received and recorded by the system. The system will then send the scoring information to the player. In addition, menu details which include game tutorials, volume settings, pause and resume functions will be provided to the players. After the players select any setting function from the menu, the game setting data will be sent to the system. After that, the system will update the game setting status for the player.

3.5 Game Design Document (GDD)

GDD is a software design document that acts as a blueprint to define the game scope and sets the general direction for this project. All the game executive summary,

detailed gameplay information, storyboard, game mechanics, game elements and also game assets of Green Genie will be included in this section.

3.5.1 Game Genre

The game genre for Green Genie is an educational and a first-person shooter genre. Players need to collect the items that are able to recycle and throw them into the correct recycle bin. Therefore, players are able to enhance their knowledge of recycling and increase their environmental awareness.

3.5.2 Game World

This game takes place in a 3D space world, where the players will need to accomplish the task for each level only they will proceed to the next level. The setting of the game takes place in Genie Land, Trash Island, Dead Tree Island and Trash Cities. Players need to clean up the trash-ridden world and fight against waste pollution which might cause a variety of diseases such as Tuberculosis, and diarrhoea.

3.5.3 Storyline

Somewhere in the universe hides a mysterious Genie Land. There is a group of Genies living in a secluded area where humans cannot see them. Jinn is one of the genies and acts as the main game character in this game. She is the descendant of the Godmother of Genie Land.

They survive on the energy emitted from The Tree of Life (Spirit Tree). One day, small genies told Jinn that sewage from the human world is slowly flowing into the ancestral river, affecting the growth of the Spirit Tree. At the same time, Godmother has gone to the spirit world for several months. They have no ideas when she will be back.

Therefore, Jinn, as the descendant of the Godmother, has to take up this responsibility to protect the human environment and also protect their Spirit Tree. Even though Jinn has no knowledge of managing all the waste in the human world, he has the power to get the Godmother's artefacts which are Plastic Blaster, Glass Blaster, Paper Blaster and GermOBlaster. Thor is the Retired Valet who had followed the Godmother for centuries. He will guide Jinn to a tunnel and teach him the basic techniques that will

be used in the human world. Then, Jinn will travel to the human world through the tunnel after he is skilled with all the necessary techniques. Jinn's journey to fight against environmental pollution begins.

3.5.4 Level Design

The game consists of one introduction scene, one tutorial scene, another five-level scene and an end scene. The introduction scene and end scene will be in Genie Land, the tutorial scene is in a tunnel that can travel to the human world while the five-level scenes will be in Trash Island, Trash Falls, Dead Tree Island and two Trash Cities with different designs respectively. The player will be given three trash blasters which are Plastic Blaster, Glass Blaster and Paper Blaster to collect and recycle the plastic bottles, glass bottles and dirty boxes or cardboard to clean up the game environments. Besides, the player will also need to use a GermOBlaster to clear and defeat the germs that are released by the waste.

3.5.5 Game Mechanics

Games are events in which players attempt to fulfil a goal within the restrictions set by the game. The restrictions of the game are more commonly referred to as game mechanics. It is the core aspect that makes a game to be more interesting and fun to play. This section will talk about all the essential components and sections that are used to develop the game.

3.5.5.1 Main Character

Players can move their game character with arrow keys or WASD keys to move in 4 directions, left, right, forward, and backwards. Besides, players can speed up the movement speed by pressing Shift together with the WASD keys. Players can press the Space bar to jump and the left button of the mouse to shoot the non-player characters (NPC), Germs which will be scattered around the game environment. Besides, the players are also able to shoot the plastic bottles, glass bottles and boxes or cardboard that have been collected to the correct recycle bin to earn points and reduce the pollution level of the environment. The task for the player is to clean the trash in the game environment and clear the Germs NPC scattered in the game environment. Since

this game application is categorized as a First-Person Shooter game genre, therefore no main character asset is needed.

3.5.5.2 Non-Player Characters (NPCs)

There are two types of enemies that will be implemented in the game environment:

- a) Germ: It acts like a normal enemy that is released by the waste pollution, therefore it is just placed statically in a few points of the game environment. Once the player defeats the Germ, the player earns 10 points.
- b) AI Germ: It acts like a smart enemy in the game and it will navigate the route to attack the player.
- c) Turtle. This is the creature that will be found on Trash Island (Level 1). The player has to clean up the plastic else the turtle will die after ingesting excessive plastic.

3.5.5.3 Reward & Punishment

In general, the physical reward of the game has the following:

- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles to the orange bin, glass bottles to the brown bin and boxes to the blue bin. Each item shot will score 10 points to the corresponding bin.
- Getting ammo box, Portal key or collectable health from the enemy after the enemy is dead.
- Getting ammo box, seeds, Portal key or collectable health from the bins if the score of the bins reached every 100 points.
- Advancing to the next level by reaching the end goal.

The more abstract reward in the game that indirectly impacts the gameplay are:

- Improving the knowledge of plastic bottles, glass bottles and boxes or cardboard waste pollution will carry bacteria which cause serious infections to humans or animals.
- Improve the knowledge of the importance of recycling which will save the animals from ingesting plastic and causing death eventually.
- Punishments that are in the game have the following:

- When the player cannot complete the tasks assigned in each level, the level fails and has to restart from the last checkpoint.
- When the player gets attacked by the Germs, the player's health bar will be deducted and the level failed and have to restart when the value of the health bar comes to zero.

3.5.5.4 Victory Condition

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game, the player is able to proceed to the next level.

3.5.5.5 Challenges

In Level 1 module, players need to collect the plastics which are also the ammo of the trash blaster in a limited time which is 900 seconds. If the player runs out of ammo, the player might die when attacked by the Germs. Besides, the player has to prevent the animals in the game environment from biting the plastic bottle. Therefore, the player has to collect the plastic bottles as fast as possible.


In Level 2, players need to collect the plastic bottles and aluminium cans and hit or recycle to Orange Bin in a limited time which is 900 seconds. At the same time, the players have to kill the germs in the environment and prevent attack by the Germs.

In Level 3, players will learn new items to recycle which are the glass bottles that have to recycle in the Brown Bin. The players have to recycle all the plastic bottles, aluminium cans and glass bottles into the correct bins in a limited time which is 900 seconds.

In Level 4, dirty boxes and cardboard will be added to the game environment. Players will learn that the paper materials objects are recycled to the Blue Bin. Hence, players have to recycle all the plastic bottles, aluminium cans, glass bottles and boxes or cardboard to the correct bins in a limited time which is 900 seconds.

In Level 5, the items that need to be recycled are the same as in Level 4. However, more Germs will be added to this level to increase the difficulty of the level.

3.5.6 Storyboard

PROJECT TITLE: Green Genie PAGE TITLE: UI - Main Menu	STORYBOARD NUMBER: 1.0
 <p><i>**The image shown is only for demonstration purposes, and does not represent the final product</i></p>	THIS PAGE CONTAINS ELEMENT BELOW: FONT: IrishGrover GRAPHICS: 2D SOUND EFFECT (SFX): <ul style="list-style-type: none">● Game Button Sfx 1 - hover● Game Button Sfx 2 - clicked MUSIC : (Calming BGM)
DESCRIPTION: <p>This is the main menu scene. There are three buttons in this scene which are the start button, the How To Play button and the exit button. If the start button is clicked, the game will start and if the players have saved files, the button will continue the progress where the player last left off.</p> <p>Clicking the “How To Play” button will open the game rules for players to view.</p> <p>Clicking the “Quit” button will close the application.</p>	

PAGE TITLE: UI - Game Rules

STORYBOARD NUMBER: 1.1



THIS PAGE CONTAINS ELEMENT BELOW:

FONT: IrishGrover

GRAPHICS: 2D UI + 3D gameplay scene

SOUND EFFECT (SFX):

- Game Button Sfx 1 - hover
- Game Button Sfx 2 - clicked

MUSIC : (Calming BGM)

ANIMATION: Game Rules panel pop-out animation

***The image shown is only for demonstration purposes, and does not represent the final product*

DESCRIPTION:

This menu shows the rules of gameplay for Green Genie.

PAGE TITLE: UI - Pause Menu

STORYBOARD NUMBER: 1.3



***The image shown is only for demonstration purposes, and does not represent the final product*

THIS PAGE CONTAINS ELEMENT BELOW:

FONT: Sarpanch-Regular

GRAPHICS: 2D UI + 3D gameplay scene

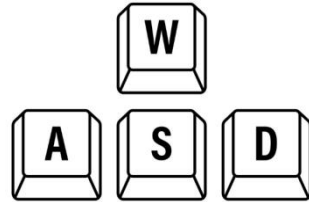
SOUND EFFECT (SFX):

- Game Button Sfx 1 - hover
- Game Button Sfx 2 - clicked

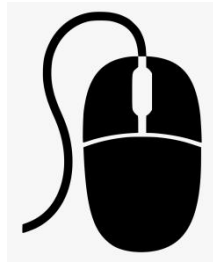
ANIMATION: Pause panel pop-out animation

DESCRIPTION:

Players are able to press the Escape (Esc) button on the keypad to pause the game. They are able to choose to continue the game, select levels that have already been unlocked or exit the game and back to the main menu.



W = Move Forward / A = Move Left / S = Move Backward / D = Move Right



Left Click = Shoot / Recycle

DESCRIPTION:

The character can walk and jump around the environment with keyboard controls. The character can shoot the enemies (Bacteria) by clicking the left key of the mouse.

<p>PAGE TITLE: Game Mechanics - Inventory System</p>	<p>STORYBOARD NUMBER: 3.0</p>
<div data-bbox="188 347 1261 914"> <p>The storyboard consists of four panels arranged in a 2x2 grid. Each panel shows a first-person view of a character's hands holding a different blaster. The top-left panel shows the 'Plastic Blaster' with 10/0 ammo. The top-right panel shows the 'Glass Blaster' with 10/0 ammo. The bottom-left panel shows the 'GermBlaster' with 10/30 ammo. The bottom-right panel shows the 'Paper Blaster' with 9/0 ammo. The background is a dark, stylized environment with some green and yellow elements.</p> </div> <p data-bbox="188 922 1400 954">**The assets used is only for demonstration purposes, and does not represent the final product</p>	<p>THIS PAGE CONTAINS ELEMENT BELOW:</p> <p>FONT: Sarpanch-Regular</p> <p>GRAPHICS: 2D UI + 3D gameplay scene</p> <p>SOUND: footsteps sound.wav, shooting sound.wav</p> <p>MUSIC : (Tense ambient BGM)</p> <p>ANIMATION: Changing weapon animation</p>
<p>DESCRIPTION:</p> <p>Players should be able to pick up the plastic bottles, glass bottles or paper boxes as ammo for the weapons (Plastic Blaster, Glass Blaster and Paper Blaster). Players should be able to scroll the mouse wheel or press the number key on the keypad to change the blaster they want to use.</p> <p>Each Blaster should have its own function:</p> <p>Pressing Key 1: Plastic Blaster</p> <p>Shoot plastic bottles to the Orange Bin and earn 10 points.</p>	

Pressing Key 2: Glass Blaster

Shoot glass bottles to Brown Bin and earn 10 points.

Pressing Key 3: Paper Blaster

Shoot paper boxes or cardboard to Blue Bin and earn 10 points.

Pressing Key 4: GermOBlaster

Shoot the enemies (Germ/AI Germ) and earn 10 points.

PAGE TITLE: Game Mechanics - Object Loot Drop

STORYBOARD NUMBER: 4.0



The left Figure shows the Germ drop ammo box after death while the right Figure shows the Germ drop Key after death.

THIS PAGE CONTAINS ELEMENT BELOW:

FONT: Sarpanch-Regular

GRAPHICS: 2D UI for health bars+ 3D gameplay scene

SOUND: footsteps sound.wav, shooting sound.wav, picking up item sound effect

MUSIC : (Tense ambient BGM)

ANIMATION: Object floating and spinning animation

DESCRIPTION:

Enemies which include Normal Germ and AI Germ have to drop ammo boxes for players to collect after they are shot until their health bar is less than zero by the players. Each ammo box contained 30 ammo for GermOBlaster. The last Germ who patrols in front of the door will drop the key for the player to enter the next level after it is dead.

PAGE TITLE: Game Mechanics - AI Navigation

STORYBOARD NUMBER: 5.0



THIS PAGE CONTAINS ELEMENT BELOW:

GRAPHICS: 2D UI for health bars+ 3D gameplay scene

SOUND: footsteps sound.wav, shooting sound.wav, enemy explodes/die sound effect

MUSIC : (Tense ambient BGM)

ANIMATION: Object floating and spinning animation, Germs and animals moving animation, Germs explode animation after being killed.



DESCRIPTION:

There are 2 crucial types of NPCs that will be using the technology of AI Navigation which are Germ to kill and Animal to protect.

Both NPCs will have their own route to different targets.


AI Germ:

AI Germ will be in every levels' game environment. The target for AI Germ is the player, they will attempt to patrol, chase and attack the player. Once the health bar of the player is attacked to less than zero, the level of the game ends.

Animal: Turtle

The game environment that has the animal with AI Navigation is in the Trash Island and the targets for turtles are plastic bottles. Players have to prevent the turtle from biting the plastics or else the turtle will die and the player will need to restart the game after the turtle's health bar is less than zero.

<p>PAGE TITLE: Game Mechanics - Health bar</p>	<p>STORYBOARD NUMBER: 6.0</p>
<div data-bbox="421 347 1176 454" data-label="Image"> </div> <div data-bbox="376 491 1220 928" data-label="Image"> </div> <p data-bbox="183 933 1406 1002"><i>**The assets used/ figure shown is only for demonstration purposes and does not represent the final product</i></p>	<p>THIS PAGE CONTAINS ELEMENT BELOW:</p> <p>FONT: Sarpanch-Regular</p> <p>GRAPHICS: 2D UI for health bars+ 3D gameplay scene</p> <p>SOUND: footsteps sound.wav, shooting sound.wav, picking up item sound effect</p> <p>MUSIC : (Tense ambient BGM)</p> <p>ANIMATION: Object floating and spinning animation</p>
<p>DESCRIPTION:</p> <p>There should be 3 types of health bars and 1 pollution level bar in the game. The three types of health bars are designed for players, enemies and animals to save respectively. The pollution level bar is designed to show players whether they have reached their goal of cleaning and recycling the waste in the game environment.</p>	

<p>PAGE TITLE: Game Mechanics - Countdown Timer</p>	<p>STORYBOARD NUMBER: 7.0</p>
	<p>THIS PAGE CONTAINS ELEMENT BELOW:</p> <p>FONT: Sarpanch-Regular</p> <p>GRAPHICS: 2D UI for health bars+ 3D gameplay scene</p> <p>SOUND: Winning.wav</p> <p>MUSIC : (Tense ambient BGM)</p> <p>ANIMATION: Level complete panel pop up</p>
<p>DESCRIPTION:</p> <p>There are tasks for the players to complete in every level. The players will be given 900 seconds to complete all the tasks. If the player cannot finish the task in the limited time, the player will need to start over the game.</p>	

3.5.7 Flowchart

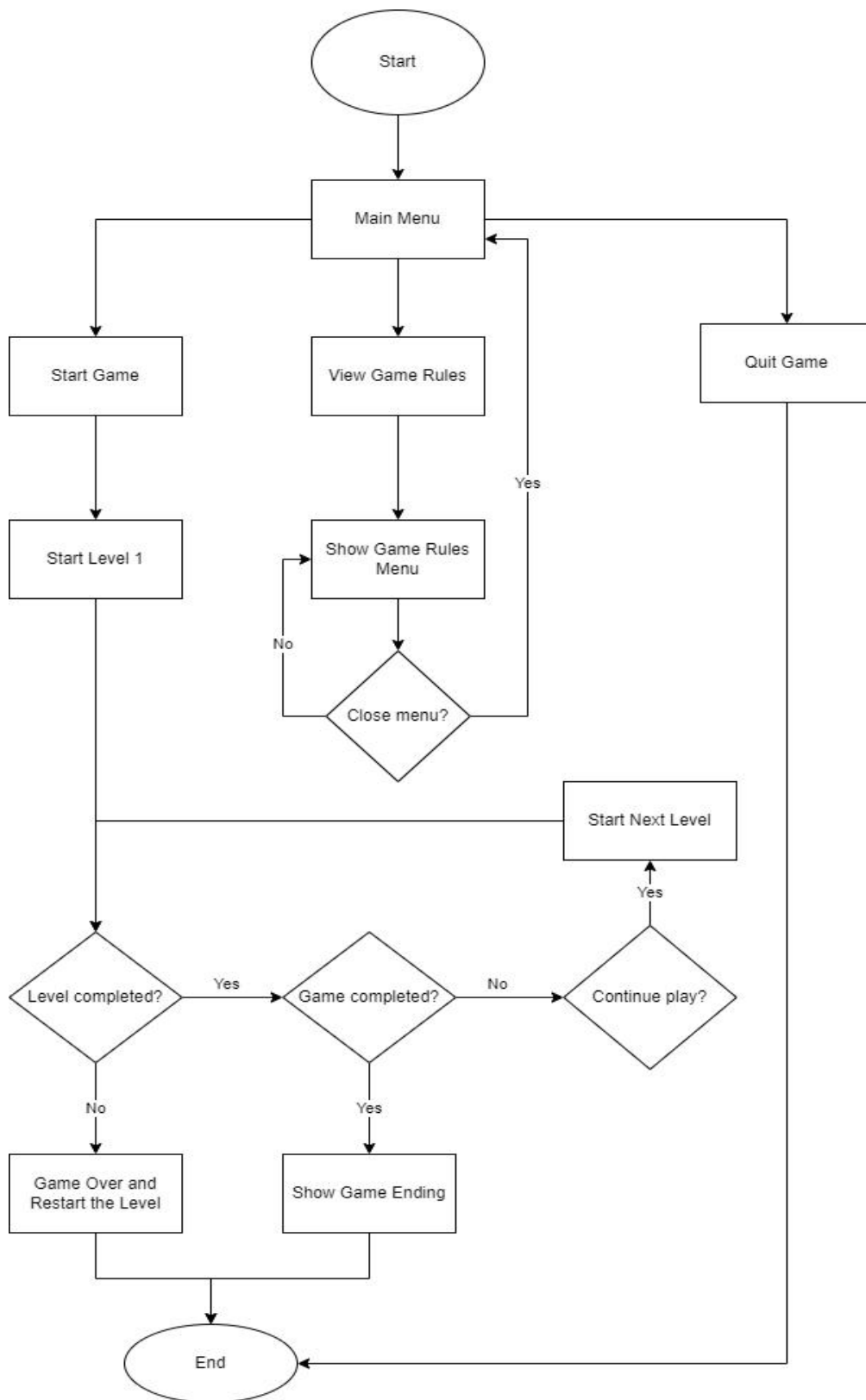


Figure 3.5 Overall Flowchart of GreenGenie

3.5.8 Game Technology

The game will be developed in 3D by using Unity Engine. Green Genie is developed in PC version therefore no network connection is required. Players are able to play the game even when offline after it is downloaded on their PC or laptops. The game will only be played by using a mouse and keyboard. The game technologies that will be used in this game are physics simulation and Artificial Techniques.

Physics Simulation (Collision, Rigid Body and Raycast)

Most of the physics simulations for all levels are the same, for example, Capsule Collider and Rigidbody are used for the player character and enemies (Germs). The Capsule Collider is composed of two half-spheres connected by a cylinder. It can be found in the Character Controller and is ideal for poles. Additionally, it can be combined with other colliders to generate unique forms. When the Colliders and Rigidbodies work together, they will bring Unity's physics to life. Normally, each object is represented by one or more geometric shapes such as boxes, spheres and capsules.

Rigidbody is a physics simulation that controls the position of an object. Gravitational forces will pull Rigidbody objects downward. If the proper Collider component is available, it will also respond to collisions with incoming objects. The Rigidbody allows the character to be affected by physics properties for example gravity, mass and velocity. As a result, the player will feel more realistic and logical while controlling the game character. Newtonian Mechanics was applied to this game of physics. For example, the rigid bodies mean the simulation's items are all absolutely solid and cannot be deformed.

Besides, ray casting will also be implemented in this scene for shooting. Raycasting is the simplest type of collision casting technology that is widely used in video game production to determine the player's or AI's line of sight, as well as the trajectory of a projectile. A raycast is a ray that travels in a specified direction and is sent out from a point in 3D or 2D space. The opponents will be damaged and have their health reduced when the raycast hits them.

Artificial Techniques (Waypoint, Navigation, Finite State Machine)

AI Pathfinding is also proposed to use in this game application. Pathfinding algorithms are commonly used as the basic foundation of any AI motion system. Pathfinding is in charge of deciding a route from one coordinate to another in the gaming world. When enemies notice the player character, they apply this AI technology to avoid barriers and pursue the player. To implement AI Pathfinding in this scenario, the A* algorithm is used in the programming part.

Table 3.3 and Table 3.4 below show the hardware and software specifications of this game application.

Table 3.3 List of Software Used in Green Genie

No.	Software	Description
1.	Adobe Photoshop CC 2019	Used to edit and design the icons, background, and story scenes of the game.
2.	Unity 3D	Used to create, design, develop and export the game files of Green Genie.
3.	Visual Studio 2019	An Integrated development environment (IDE) that is used in Unity for scripting in C# Language.
4.	Google Docs	Use for documentation.

Table 3.4 List of Hardware Used in Green Genie

No.	Software	Description
1.	Asus TUF Gaming FX505DD Laptop	The core device that is used for downloading software, editing assets, designing levels and building the game application.

3.6 Data Design

A multimedia database management system (DBMS) is used for this project development to facilitate the construction, storage, access, querying, and control of multimedia databases. It is a structured collection of multimedia data elements such as text, images, graphics, animation, and audio. A multimedia DBMS must provide multimedia data modelling, multimedia object storage, multimedia indexing, retrieval, browsing, and multimedia query support. This subtopic provides a generic Unity framework used for this game project.

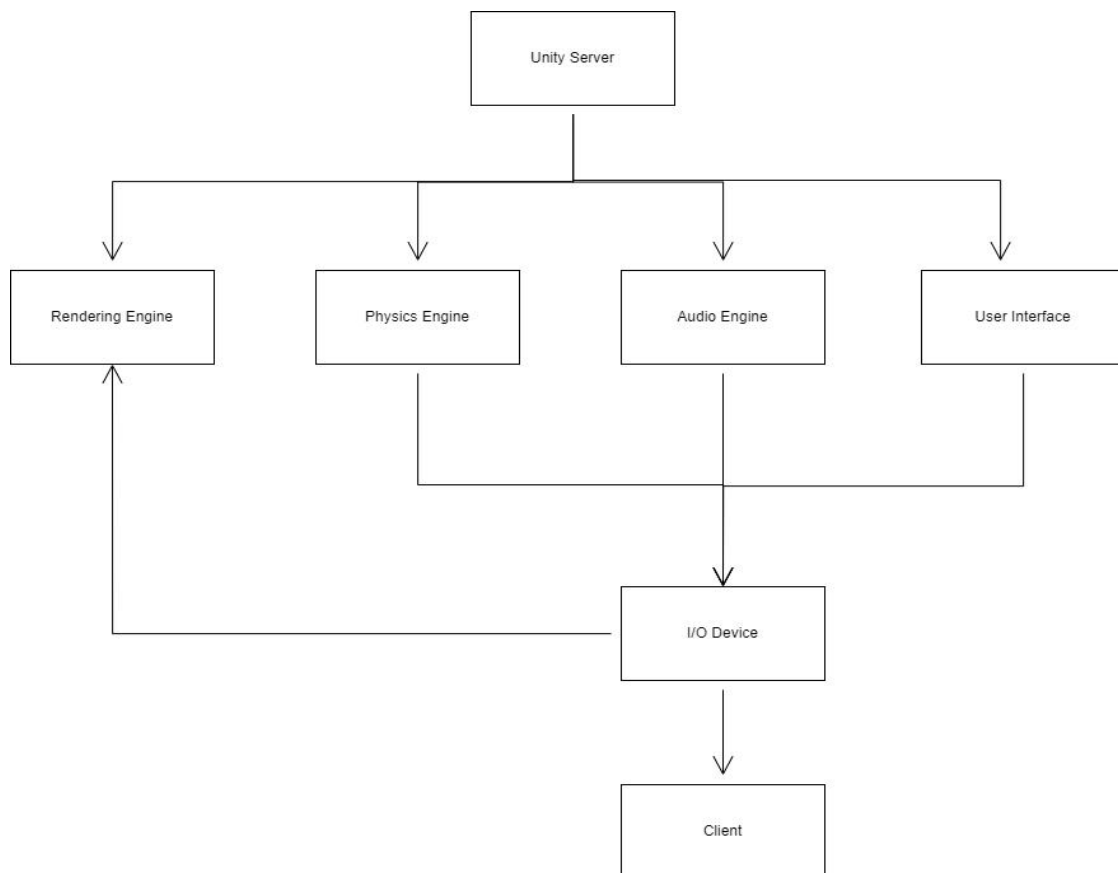


Figure 3.6 Data Design for developing GreenGenie

Unity game Engine is the heart of developing a game and consists of several components. Rendering Engine, Physics Engine, Audio Engine and Scene Manager (User Interface Engine) are the main components of a common architecture of a game engine.

Rendering Engine also known as the graphic engine works with an application program to draw graphics on the computer display. This engine involves objects

transform, lighting calculation, viewing transform, projection and rasterization. For example, the lightning and particle effects of Green Genie will be using this Rendering Engine.

Unity Physics Engine allows 3D objects to be controlled by real-world forces like gravity, velocity, and acceleration. For example, the key physics include in the engine have collision detection, trigger detection and rigid body dynamics and also being used in Green Genie project for the player controller and shooting system.

Audio/Sound Engine supports most major audio file formats and includes extensive tools for playing sounds in 3D space, with optional effects such as echo and filtering. Unity demands that sounds come from Audio Sources attached to the game objects. The audio audios emitted is then picked up by an Audio Listener linked to another game object, usually the main camera. Unity can then simulate and playback the effects of a source's distance and location from the listener object. For example, the sound effect such as collecting items and background music in Green Genie has to use Audio Engine.

User Interface Engine (Unity UI) is a crucial component for developing user interfaces for games. The capabilities of Unity UI include generating Canvas, Button, Image, Text, Slider, Sprites, moving and animating objects, defining user interactions, and automatically scaling layouts. For example, all the user interfaces in Green Genie such as main menu panel, pause menu panel, mission panel, weapon information panel and all the texts shown need the use of User Interface Engine.

These are the crucial components of a game development framework. Without one of the multimedia elements, the functionality and experience of the game will be affected.

3.7 Testing Plan

The testing process is a very crucial part of the game development to test its functionality and usability. It acts as a final defence to check for any errors and room for improvement of the game application. The technique that is planned to be used for the test process of GreenGenie is User Acceptance Test (UAT). There are two types of testing that will test in the testing process which are the Alpha test and the Beta test.

The Alpha test is about testing the functionality of the game while the Beta test is about user testing in which the user is chosen to play the game and get the user feedback through Google Forms.

3.7.1 Alpha Test

The tables shown below are the set of test cases that are planned to test the functionality of the game after the development phase.

Table 3.5 Test Case of Main Menu

Test Case 1: Main Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Result
1	Start the game from the first level.	Click on the start button	The first level of the game will be loaded.	PASS/FAIL
2	Go to the game instruction menu	Click on the How To Play button.	The game instruction menu will be popped out.	PASS/FAIL
3	Quit the game application	Click on the quit button	The game will be closed.	PASS/FAIL

Table 3.6 Test Case of How To Play Menu

Test Case 2: How To Play Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Result
1	Show the player the instruction for gameplay	Click on the How To Play button.	The game instruction menu can be viewed.	PASS/FAIL
2	Go back to the main menu.	Click on the back button.	The main menu will show up back.	PASS/FAIL

Table 3.7 Test Case of the Gameplay

Test Case 3: Gameplay				
No.	Functionality	Pre-Condition	Expected Outcome	Result
1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	PASS/FAIL
2	Player character can jump	Press the spacebar on the keypad.	The character can jump.	PASS/FAIL
3	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	PASS/FAIL
4	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	PASS/FAIL
5	Animals in the gameplay can eat plastics and die after the health bar is reduced to zero.	Animals will auto-move in their path that has been set in the system when the level starts.	Animals in the gameplay can eat plastics and die after the health bar is reduced to zero.	PASS/FAIL
6	The Brown Germ can attack the player character.	The player character goes near the Brown Germ.	The Brown Germs attack the player character.	PASS/FAIL
7	Germs in the gameplay can drop ammo box or key.	The player character shoots down the germ.	Germs in the gameplay drop ammo box or key.	PASS/FAIL
8	Points can be added.	The Player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	PASS/FAIL

Table 3.8 Test Case of Pause Menu

Test Case 4: Pause Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Result
1	Resume the game.	Players press the pause button (Esc key) when the game is paused.	The game will be resumed when the pause button is pressed when the game is pausing. The pause menu will be hidden away.	PASS/FAIL
2	Restart the level.	Players press the restart key (R key) or button.	The level will be restarted from the start.	PASS/FAIL
3	Load the main menu	Players must first pause the game and then press the main menu button.	The current progress of the level will be lost and the game brings the player back to the main menu scene.	PASS/FAIL

3.7.2 Beta Test

The beta testing phase is getting the users' feedback and therefore the game is planned to upload on a web page so that the reviewers can download and test playing the game. A google form about the functionality of the game will be distributed to the reviewers and get feedback from them.

3.8 Potential Use of Proposed Solution

The goal of this proposed game application, GreenGenie, is to prompt the young generation to think about the importance of taking care of our environment. Recycling is one of the ways to protect our environment from waste pollution.

This proposed game uses an indirect way to send messages to the players about the waste pollution that can bring infections due to the spread of harmful bacteria or germs in the surroundings and also threatens the animals' life, especially aquatic life. Based on the research of the Institute for the Study of Human Knowledge, there are at least one million seabirds and ten thousand marine animals dying every year due to ingesting plastics which are not digestible of course. However, there are still many people who do not realize the act of simply littering plastic bottles or plastic bags can even cause a small creature to die.

Therefore, this project attempts to raise environmental awareness among children nowadays. It is critical to begin teaching kids habits at an early age, especially given that children form habits by the age of nine. Both bad and good habits can be difficult to break. The earlier we start teaching good behaviour, the more likely they will persist for years to come

CHAPTER 4

RESULTS AND DISCUSSION



4.1 Introduction

This chapter discusses the development and implementation process of Green Genie, the 3D educational and first-person shooter game which aims to increase awareness of environmental issues. Topics that will be discussed in this section include game design, game interface, coding scripts, and tools used to develop the game.

4.2 Project Setup

There are a few steps that must be completed before beginning the actual development of the game. Software installation, version control, and package imports are all part of the setup. Table 4.1 shows all of the software and tools required for this project.

Table 4.1 Software to setup during the development of GreenGenie

Software	Description
 Adobe Photoshop CC 2019	Graphics editor that used to design sprites, background, and story scenes of the game.
 Unity 3D	A game engine used to combine all of the coding, graphics, and audio into a gaming application. This project's version number is 2021.3.10f1.



 Visual Studio 2019	An integrated development environment (IDE) for scripting in the C# language that is utilised in Unity.
 Blender	An open-source 3D computer graphics software tool used for creating models and animation.

Table 4.2 Essential Unity Packages Installed

Packages	Description
Shader Graph	The Shader Graph package extends Unity with a visual Shader editing tool. Instead of writing code, the shaders are able to construct visually. Unique graph properties can be implemented using specific render processes.
Unity Terrain Editor	An environment design tool which offers a set of Terrain tools that allows for the creation of realistic terrains.
Universal Renderer Pipeline (URP)	A prebuilt scriptable render pipeline from Unity that provides artist-friendly workflows for creating optimised graphics across multiple platforms.

4.3 Implementation Process

This section will describe the process of capturing all of the stages involved in developing the Green Genie game application.

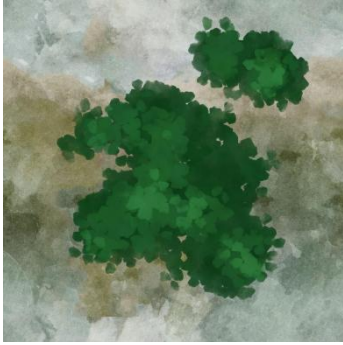
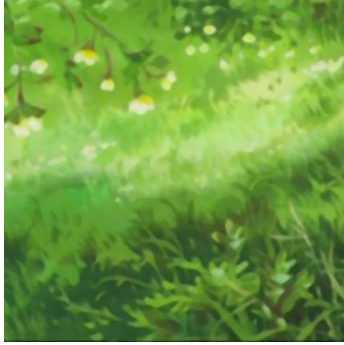
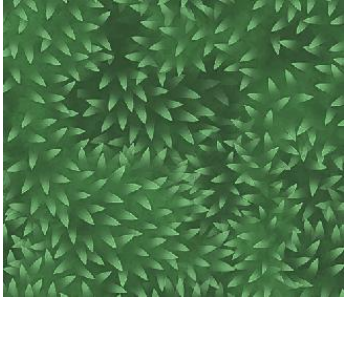
4.3.1 Designing Graphic and Multimedia Contents

Before the developing process, all of the necessary graphics and multimedia content must be prepared. This game application's graphic and multimedia elements are as follows:

4.3.1.1 Images

The UI interface, buttons, panels and models usually need the image to aid immersion in the game since the image can be used as a texture or material in Unity. Some of the objects in the game application will require images to be included as materials in order to seem more realistic and attractive. Table 4.3 below shows the images that were designed or edited from Photoshop and will be imported into Unity as materials or textures. Some of the images will also be the information on the 3D models. For example, the recycle logo will be imported into Blender as a plane and pasted on the surface of recycle bin models. All the images will be converted into PNG format.

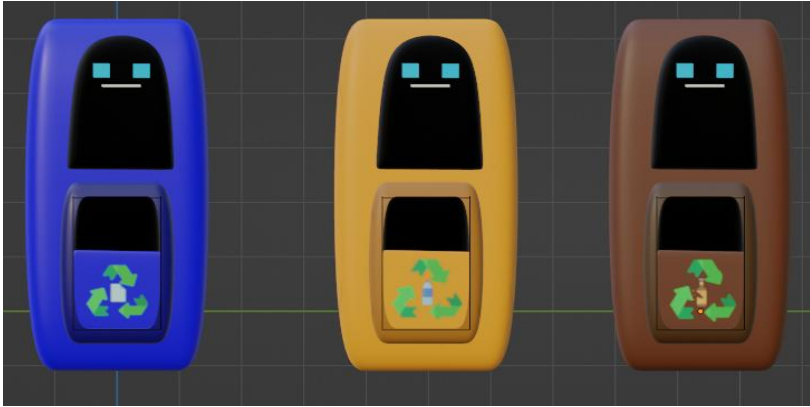
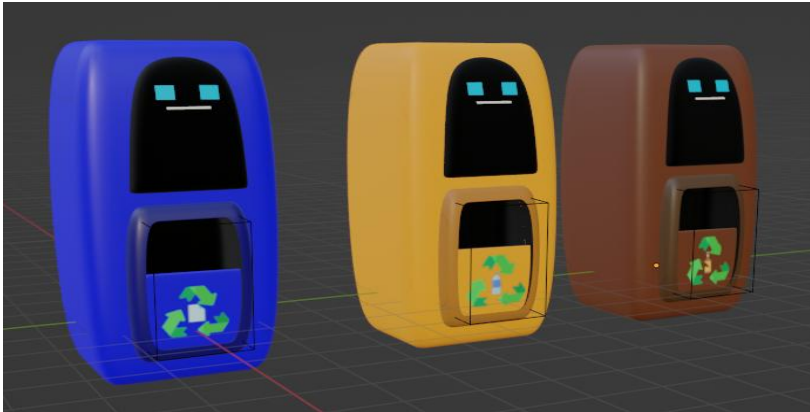
Table 4.3 Images that are imported into Unity or Blender as a plane

		
This is the image designed as material to decorate and assign to the rock model.	This is the image designed as material to decorate the ground of the game environment.	This is the image that will be converted to grass texture in the game environment.

4.3.1.2 3D Models

Most of the 3D Models in the game scene will import from the Unity Asset Store. However, there are few 3D models in the game scenes that will be implemented by using Blender. Table 4.4 shows the 3D models that were created on my own and included in the game project.

Table 4.4 The details of 3D models included in the game project.

<p>1. The 3 Types of Recycle Bins: Paper Recycle Bin (Blue), Bin (Orange) and Chocolate Bin.</p> <p><u>Front View</u></p>  <p><u>Perspective View</u></p> 
<p>These are the models that were created by using Blender. The modifier properties to build the models include the Bool Tool - Difference, and also the subdivision surface to smooth the models. Figure 4.1 shows the modifier properties of the recycle bin models.</p>

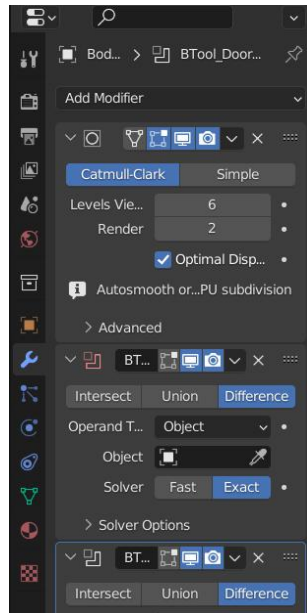


Figure 4.1 Modifier properties of the recycle bin models.

The Shader Editor is also being used to create materials and colour for 3D models. Then, fill colour and paint the models using Texture Paint,



There are many useful features that can be used such as Soften, Smear, Fill and Draw.

2. Fences for planting



Figure 4.2 Fence model

This is the model downloaded from Unity Asset Store and it will be used to make a planting area for the player. Figure 4.3 shows the final output of the planting area.



Figure 4.3 Planting area

3. Waterfall

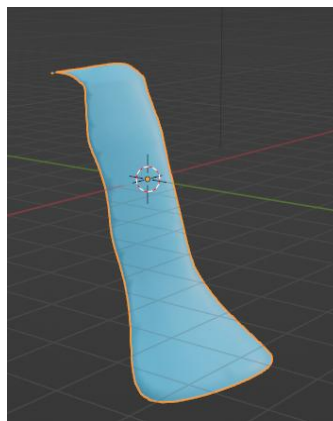


Figure 4.3 Waterfall shape model

To make the waterfall with the shader graph effect, the waterfall shape has to be modeled and assigned to the material under the render section in the particle system. Figure 4.4 shows the waterfall material assigned.

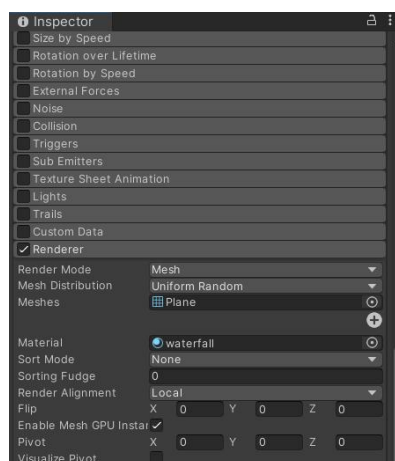


Figure 4.4 Assigning waterfall material into the particle system under the render section.

4.3.1.3 Terrain

The game environment for each scene will be created using the Unity Terrain Editor. It is a tool to create realistic terrains. There is a set of Terrain Tools provided in the editor which helps to design the optimized landscapes efficiently. There are only few scenes using the Terrain which are Introduction Scene, End Scene, Tutorial Scene, Level 1 Scene, Level 2 Scene and Level 3 Scene. The environment in Levels 4 and 5 are using 3D assets downloaded from Unity Asset Store which will be discussed in the next section. (Topic 4.3.3 Game Environment Setup). Table 4.5 depicts the details and design method for creating the terrains.

Table 4.5 The details and designing method for creating terrains.

1. Terrain for Introduction and End Scene: Genie Land

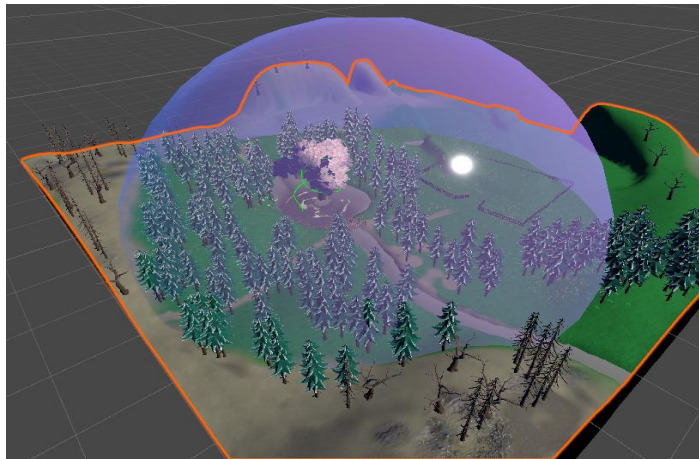



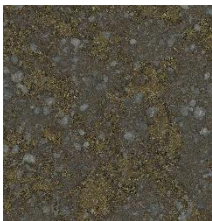

Figure 4.5 Overview of Genie Land Terrain



Figure 4.6 Spirit Tree of Green Genies

This terrain is only used for shooting animation for the introduction story of the game, therefore there are only certain parts that needed to capture have the trees. This is to reduce the glitches issues if there are too many assets that need to be rendered. The purple hemisphere is the boundary between the human world and Genie Land.

Material Used for Paint Terrain

	 		
Rock material	Grass materials	Soil material	Ground material

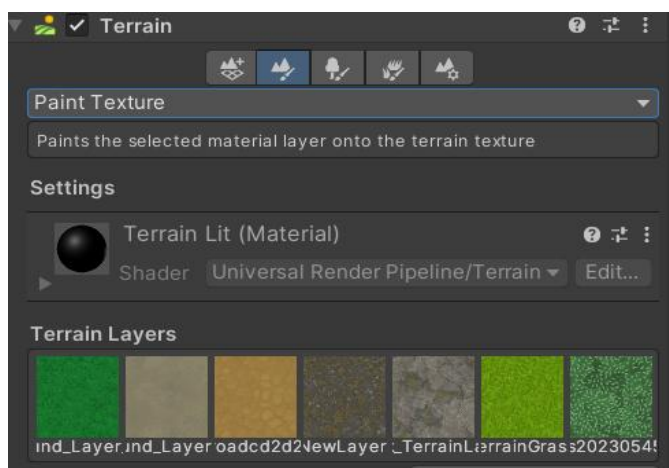


Figure 4.6 Paint Texture Setting in Terrain Editor.

Figure 4.6 shows the materials that were used to design and paint the terrain.

Material Used for Paint Trees

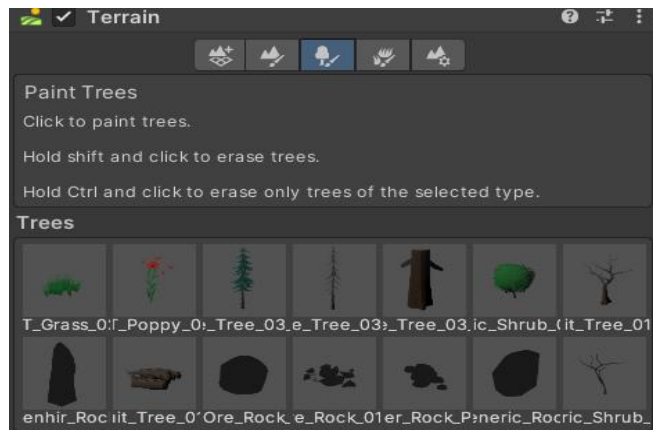
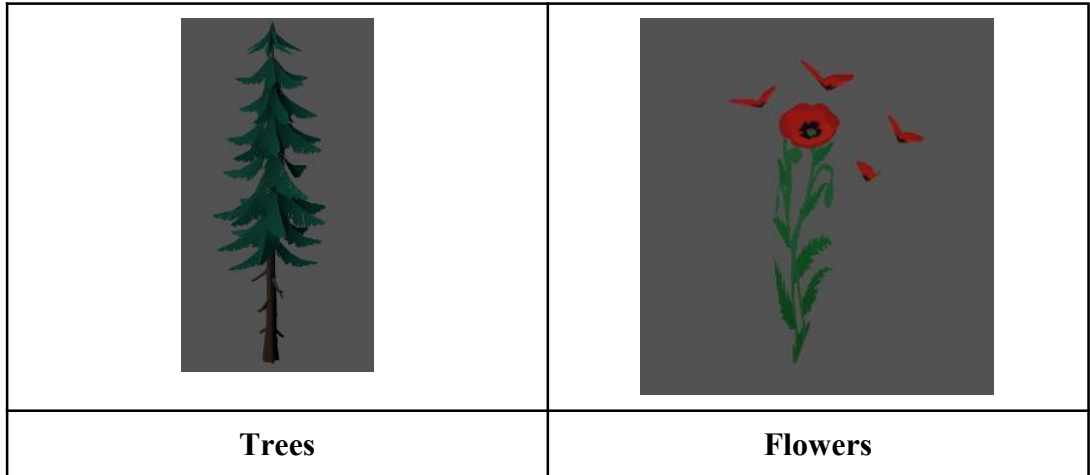


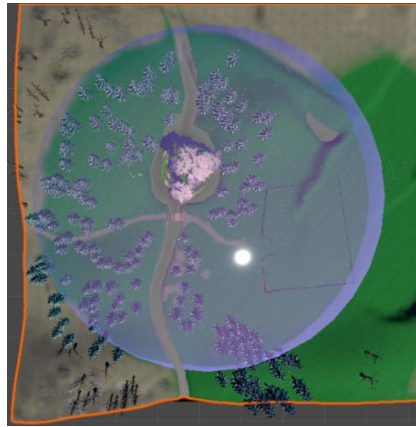
Figure 4.7 Paint Trees Setting in Terrain Editor.



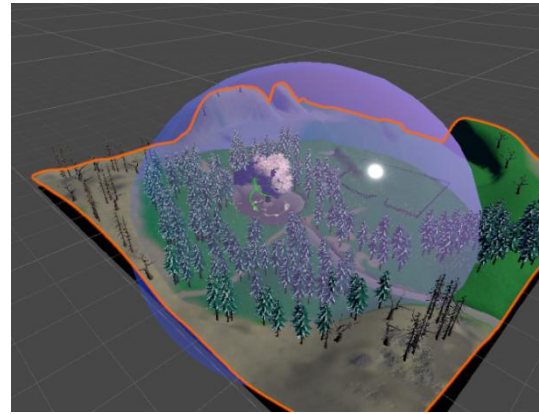
Figure 4.8 The area that wants to paint tree

Figure 4.8 shows the prefab assets that were used to decorate the terrain while Figure shows the area that wants to paint trees. The size of the brush used to paint trees can be adjustable.

Results of Terrain for Tropical Forest Scene



Top View



Perspective View

Figure 4.9 Results of Terrain after adding all the materials.

2. Terrain for Tutorial Level: Tunnel to Human World

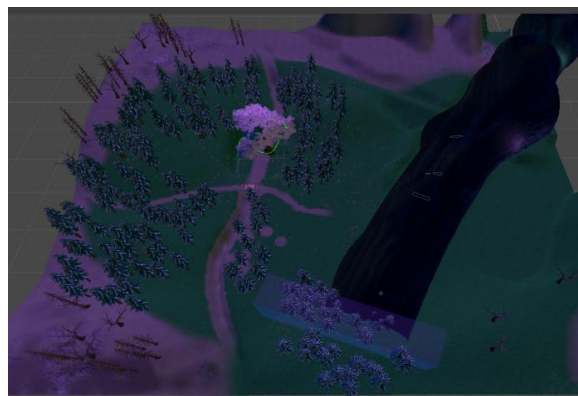


Figure 4.10 Results of Terrain after adding all the materials.

The terrain for Tutorial Level is the same as Introduction Scene, just a 3D tunnel model is added to the scene. Therefore, the materials used to paint the terrain are all the same as Introduction Scene as mentioned above.

3. Terrain for Level 1: Trash Island

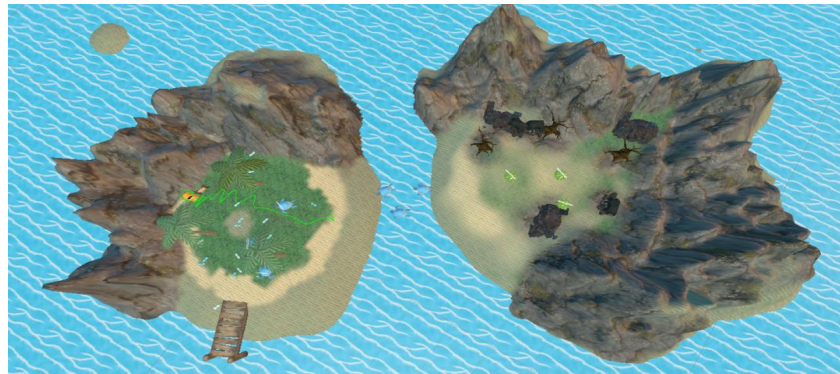

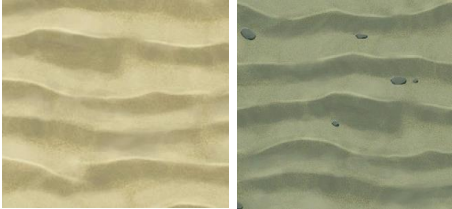



Figure 4.11 Overview of Trash Island Terrain

There will be mountains, sea, sand painted on this Terrain.

Material Used for Paint Terrain

		
<p>Rock material</p>	<p>Sand materials</p>	<p>Grass materials</p>

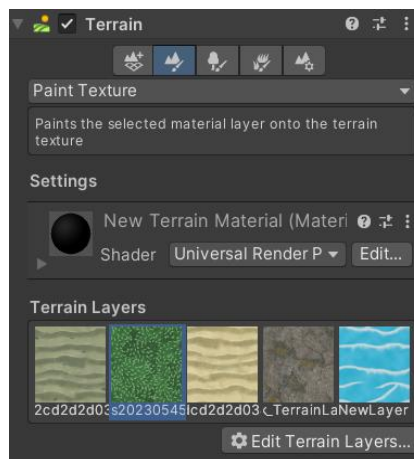
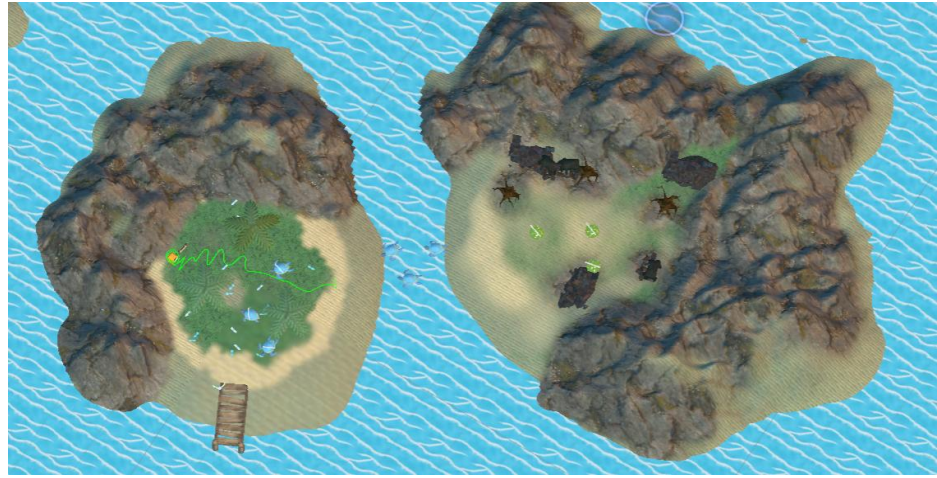


Figure 4.12 Paint Texture Setting in Terrain Editor.

Figure 4.12 shows the materials that were used to design and paint the terrain.

Results of Terrain for Trash Island Scene



Top View



Perspective View

Figure 4.13 Results of Terrain after adding all the materials.

4. Terrain for Level 2: Trash Fall

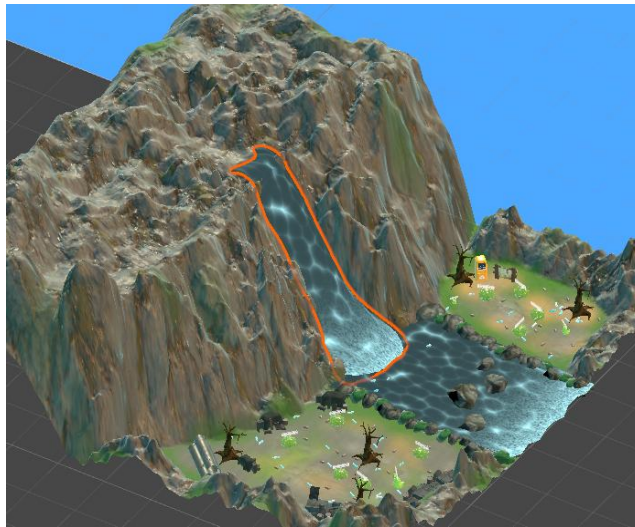

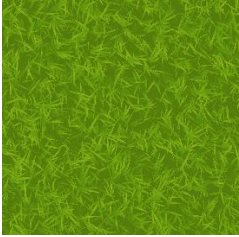
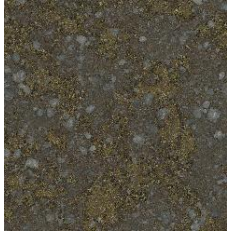



Figure 4.14 Overview of Trash Fall Terrain

There will be mountains, rocks, ground and grasses painted on this Terrain.

Material Used for Paint Terrain

			
Rock material	Grass materials	Soil material	Ground material

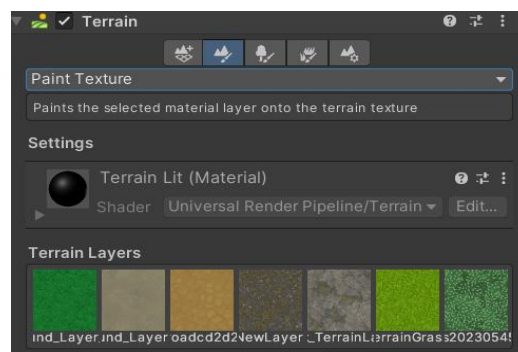


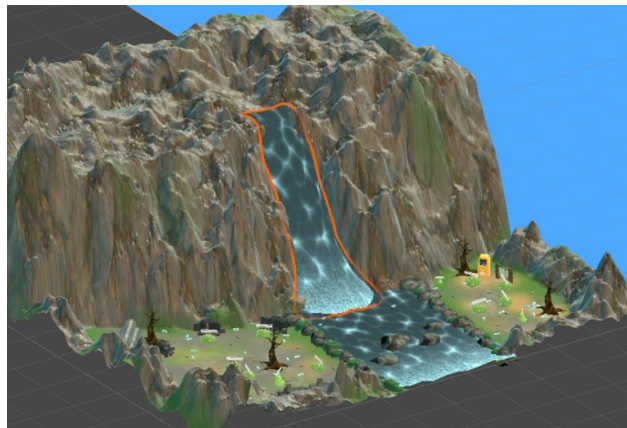
Figure 4.15 Paint Texture Setting in Terrain Editor.

Figure 4.15 shows the materials that were used to design and paint the terrain.

Results of Terrain for Trash Fall Scene



Top View



Perspective View


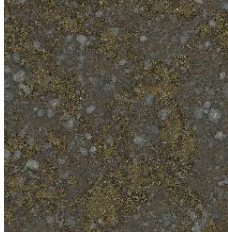
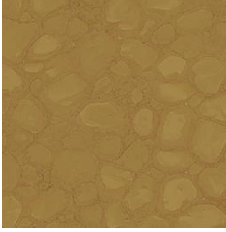
5. Terrain for Level 3: Dead Tree Island



Figure 4.16 Overview of Dead Tree Island Terrain

There will be mountains and two type of sand painted on this Terrain.

Material Used for Paint Terrain

		
Rock material	Soil material	Ground material

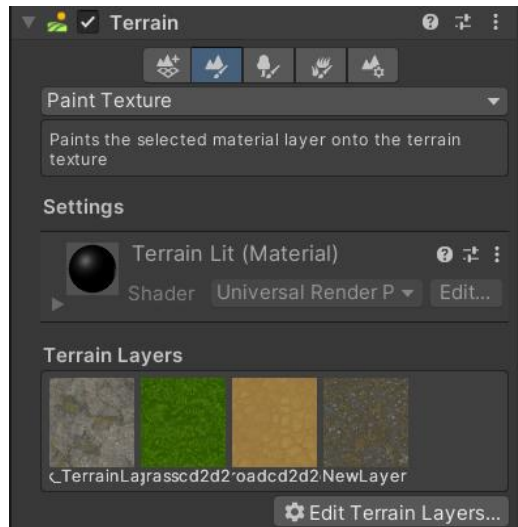


Figure 4.17 Paint Texture Setting in Terrain Editor.

Figure 4.17 shows the materials that were used to design and paint the terrain.

Material Used for Paint Grass



Grass

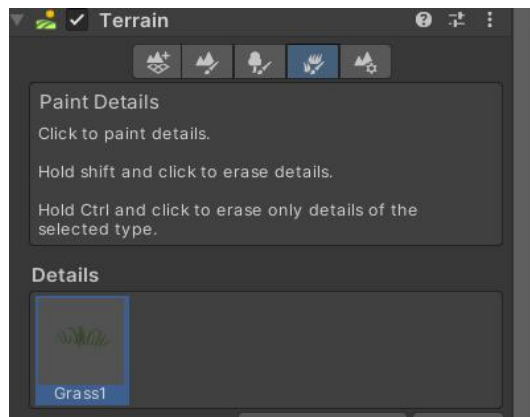


Figure 4.18 Paint Grass Setting in Terrain Editor.

Results of Terrain for Dead Tree Island Scene



Top View



Perspective View

Figure 4.19 Results of Terrain after adding all the materials.

4.3.1.4 Shader Graph

There are a total of two Unlit shader graphs created for making the waterfall effect which are the waterfall splashing down effect and ripple effects.



Figure 4.20 Waterfall for Scene 3 Game Environment

1. Waterfall splashing down Shader Graph (named as *waterfall.shader* in Unity)

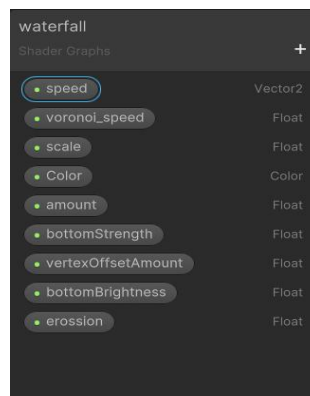


Figure 4.21 Property of Waterfall Shader Graph

The Figure above shows the essential property that has to be included in building the Shader Graph. The amount is to control the speed, brightness and strength of the waterfall's splash.

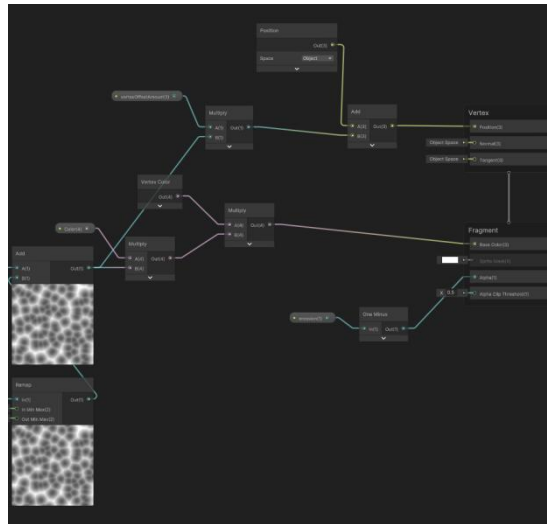
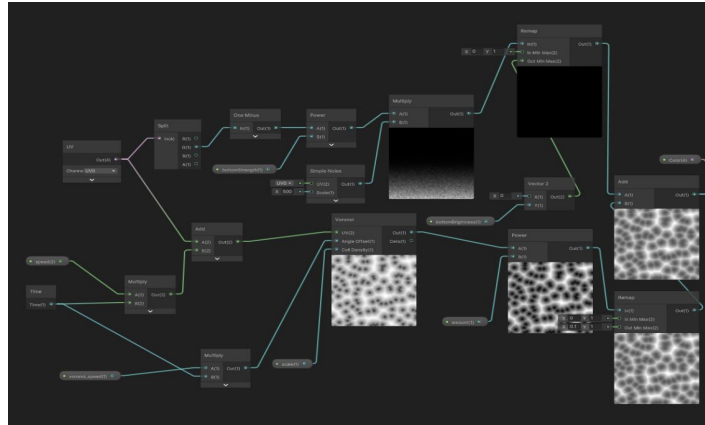


Figure 4.22 Screenshots of Shader Graph built for Waterfall Splashing Effect

The Figure shows that a voronoi node is added to the graph shader because of the scale, speed and amount of these cells resulting in noise which can make the model more realistic.

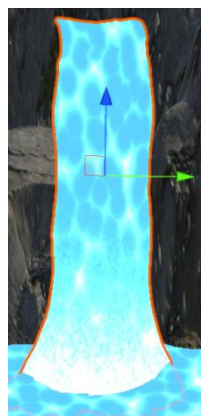


Figure 4.23 Result of WaterfallShader

2. Ripples Shader Graph (named as scrollShader.shader in Unity)

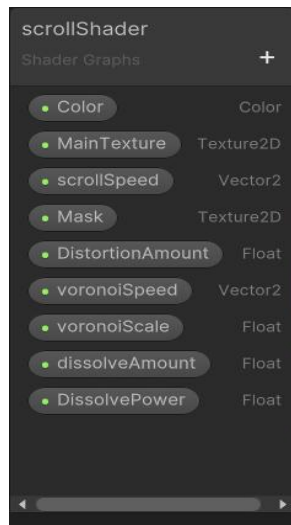


Figure 4.24 Property of ScrollShader Graph

The Figure above shows the essential property that has to be included in building the Shader Graph. The amount and scale are also the same as waterfall.shader, used to control the speed of scroll and voronoi, brightness and strength of the waterfall's splash. However, a main texture is added for this shader to show the ripple shape.



Figure 4.25 Ripple Lines

The figure above depicts the wavy lines created in Photoshop and imported into Unity as a 2D texture.

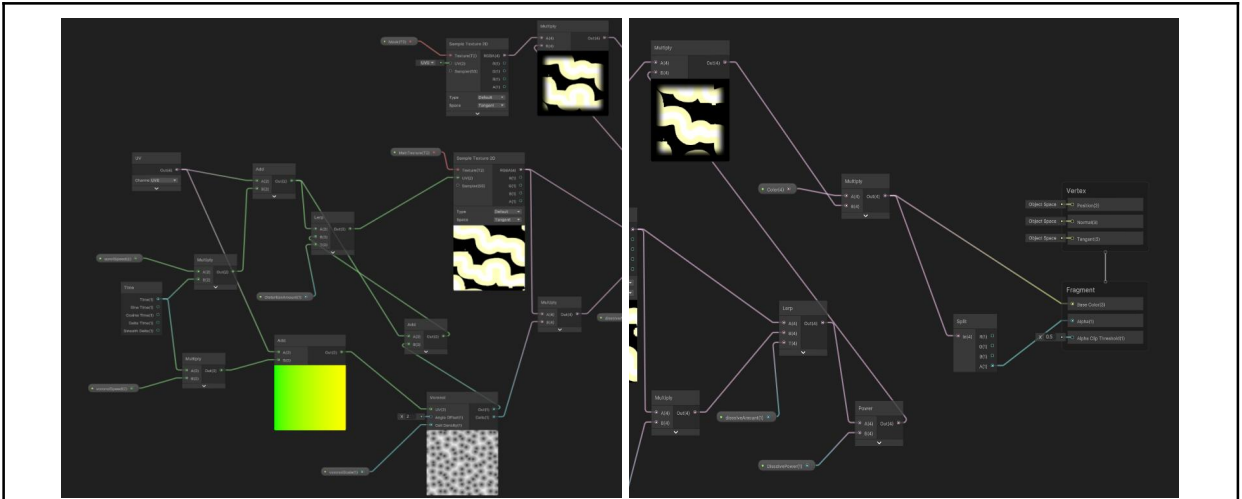


Figure 4.26 Screenshots of Shader Graph built for Scroll Shader Effect for ripples

The Figure 4.26 shows that voronoi node and wave texture is added to the graph shader. The scale, speed and amount of the voronoi cells result in noise which can make the model more realistic while the texture can form the circular shape for the ripples.



Figure 4.27 Result of ScrollShader

4.3.1.5 Audio

The audio used is imported from the Unity Asset Store. The figure below shows all the audio that will be used throughout the game.

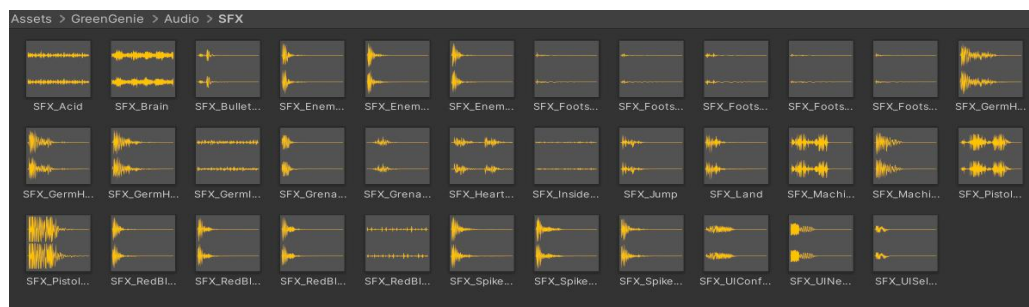


Figure 4.28 Audios in the game

4.3.1.6 Standard Text and Text Mesh Pro

Text is an essential element in a game application to show information such as the game title, game rules, score text and so on.



Figure 4.29 Standard text in the game

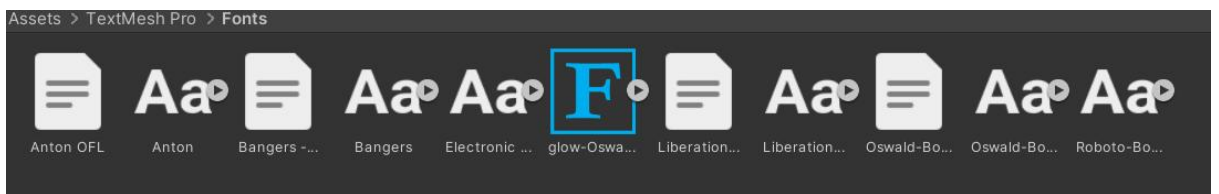


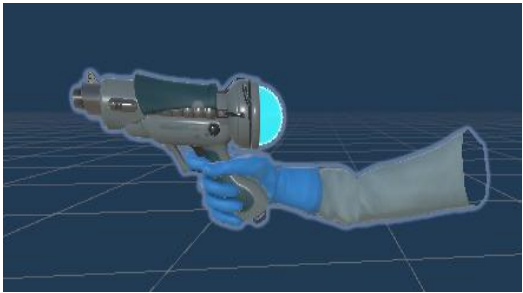

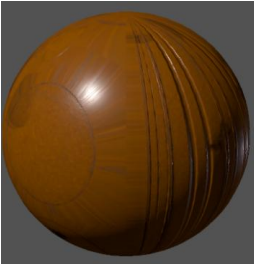
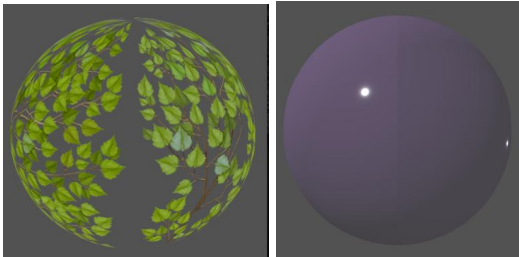
Figure 4.30 Text mesh pro in the game

4.3.2 Game Characters Control

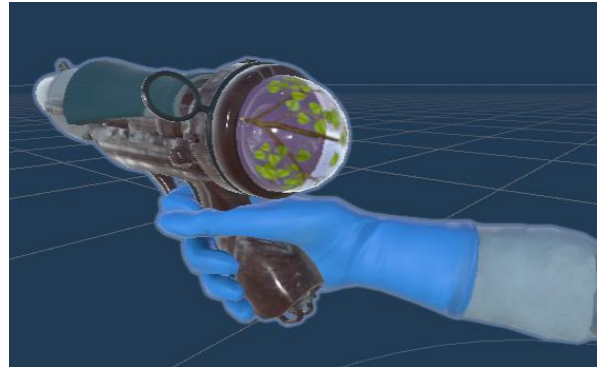
4.3.2.1 Main Character Setup

Jinn is the main character in this game application. Since this is a first-person shooter game application, therefore only weapons and hand-holding the weapons needed. The 3D models which include all the weapons and hands of the main character are imported from Unity Asset Store. Table 4.6 shows the results of editing the material of the original 3D weapon model.

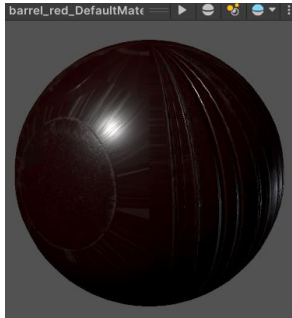
Table 4.6 Before and after editing the original 3D models

Before the material editing of the original 3D model	After the material editing of the original 3D model
1. Plastic Blaster	
	
<u>Materials used for the editing process:</u>	
	
Body material	Glass materials
<p>Description: The body of the Plastic/Can Blaster changed to Orange material to allow the player more familiar with the recycling knowledge on plastic bottles and aluminium cans are recycled to the Orange Bin.</p>	

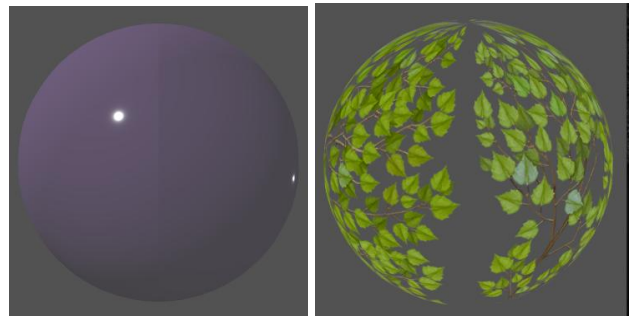
2. Glass Blaster



Materials used for the editing process:



Body material

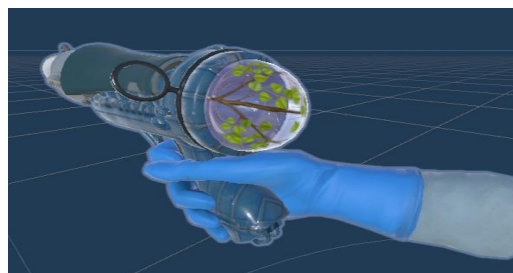
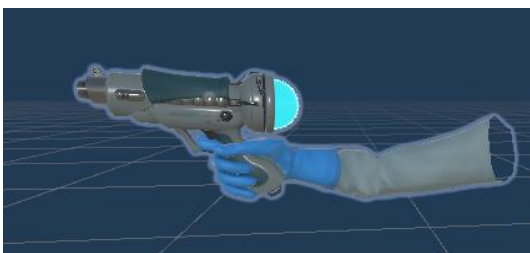


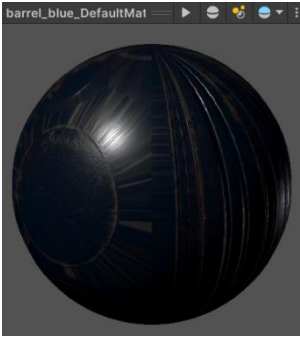
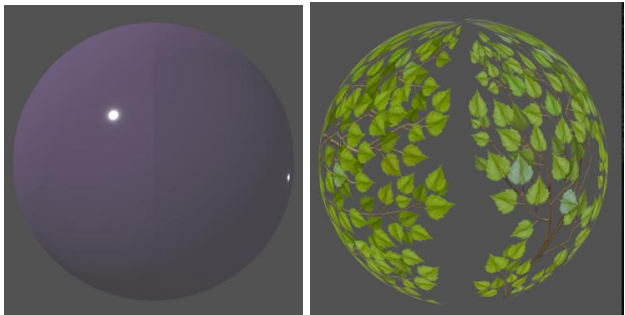
Glass materials

Description:

The body of the Glass Blaster changed to Chocolate red material to allow the player more familiar with the recycling knowledge on glass bottles are recycled into the Brown Bin.

3. Paper Blaster



<u>Materials used for the editing process:</u>	
	
Body material	Glass materials
<p>Description:</p> <p>The body of the Paper Blaster changed to Blue material to allow the player more familiar with the recycling knowledge of paper materials such as boxes and cardboard are recycled to the Blue Bin.</p>	

Main Player Controller

The main player character has 3 basic animations which are idle, walk and run. Each level of the player character has the same motion animation. The movement of the player character is controlled using the basic keys on the computer keyboard, with the W, A, S, and D keys in place of the arrow keys. With the WASD keys, W is moving forward, A is moving leftward, S is moving backwards, and D is moving rightward. The space bar is the jump control key in this game. Holding down the Shift key and the move keys at the same time could control the player character to speed up. Other animation states of the player character include reload and fire.

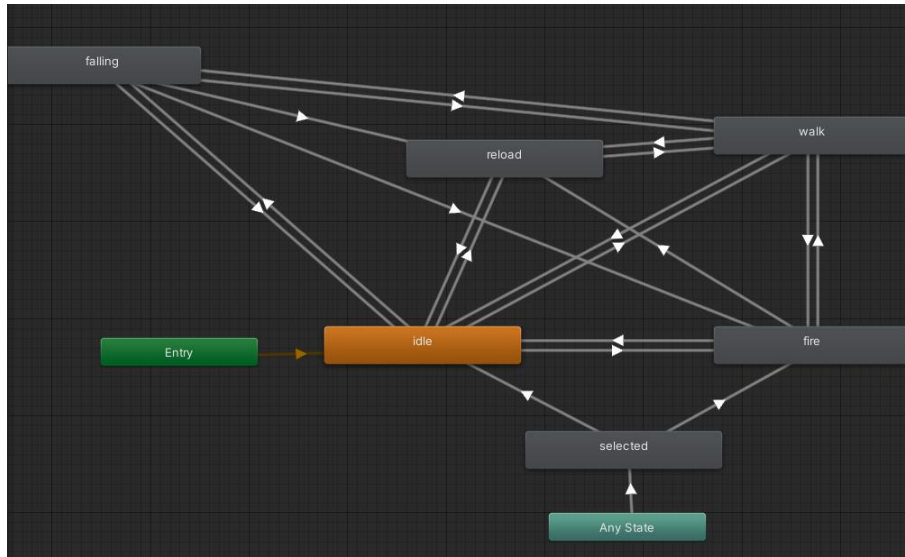
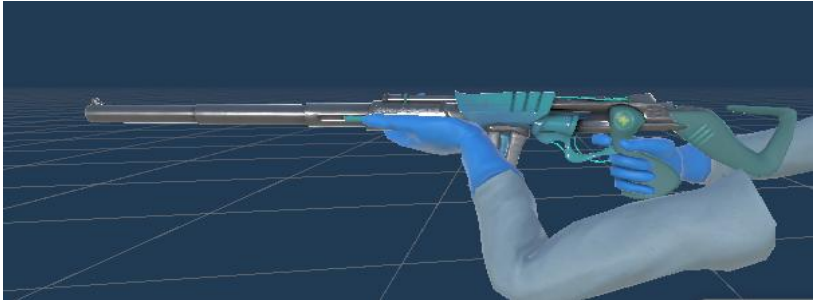

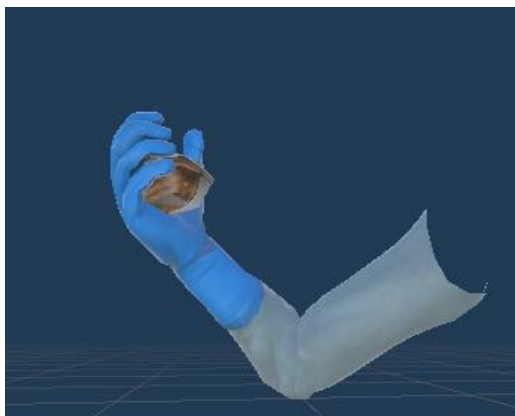


Figure 4.31 Screenshot of the Animator of the main character in Unity.

Table 4.7 below shows the arranged screenshots of the player character’s animation states in Unity.

Table 4.7 Screenshots of the player character’s animation states

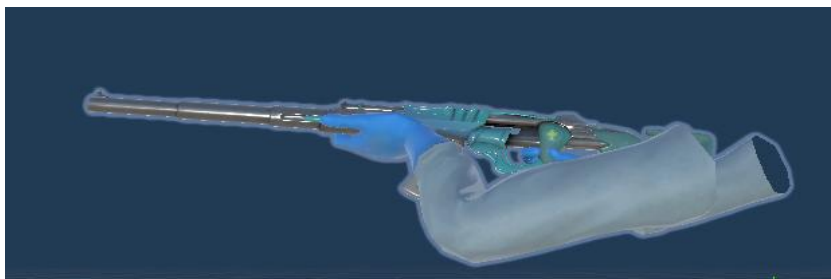
Animation States	Basic Animation
Idle	 <p data-bbox="786 1480 1225 1518"><i>Idle Animation for GermOBlaster</i></p>  <p data-bbox="560 1868 1453 1942"><i>Idle Animation for Plastic Blaster (Same as Glass Blaster and Paper Blaster)</i></p>



Idle Animation for Seed Projectile

Walk

The walk animation is not too obvious, the walk state is just a very slight shaking of the arms.

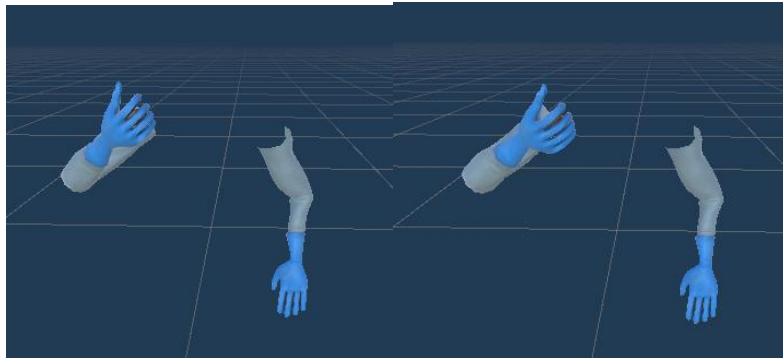


Walk Animation for GermOBlaster



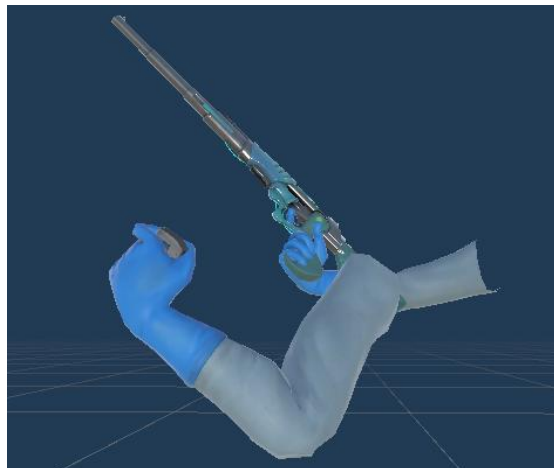
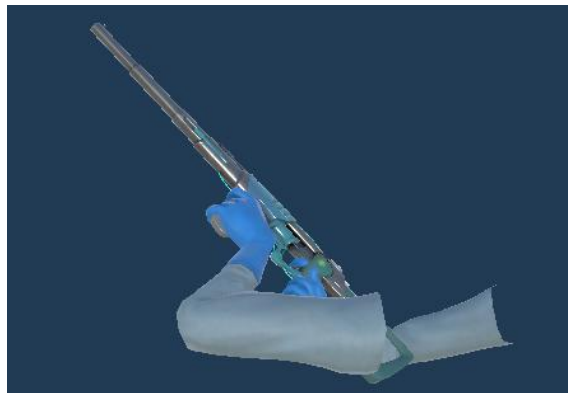


Walk Animation for Plastic Blaster (Same as Glass Blaster and Paper Blaster)



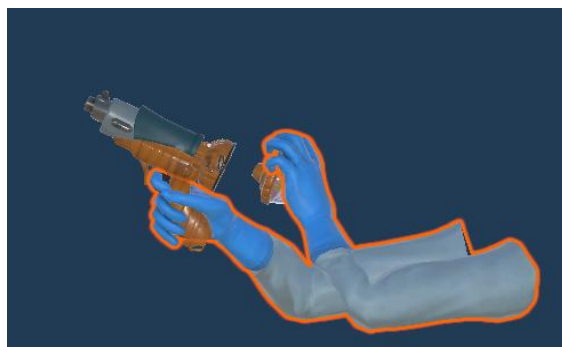
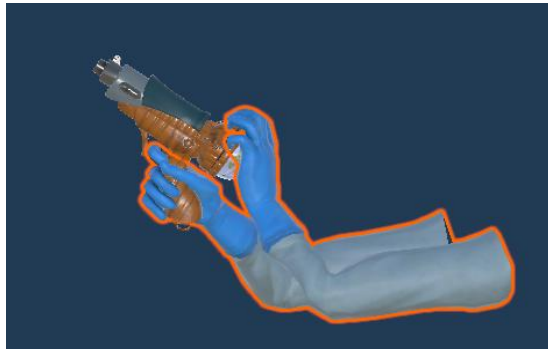
Walk Animation for Seed Projectile

Reload

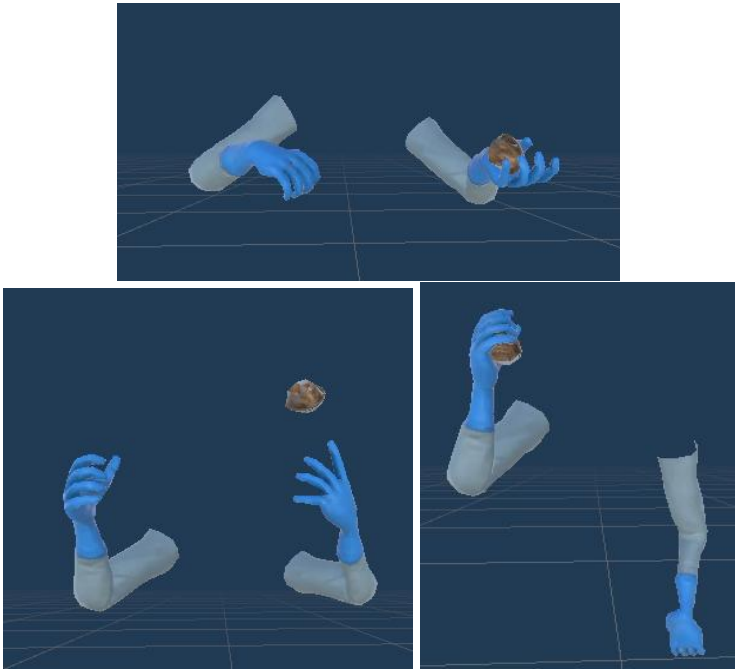
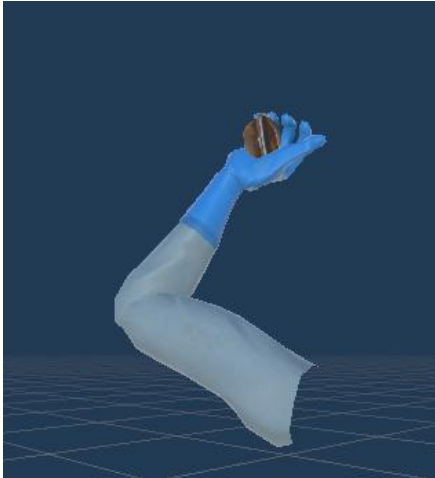
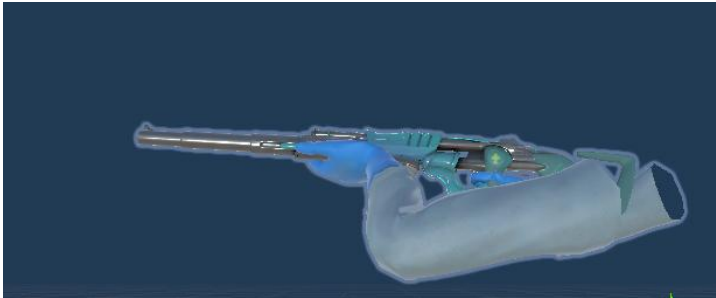


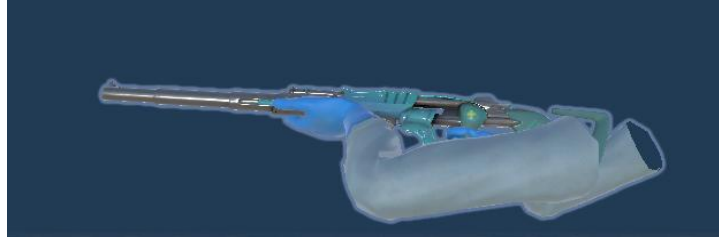


Reload Animation for GermOBlaster

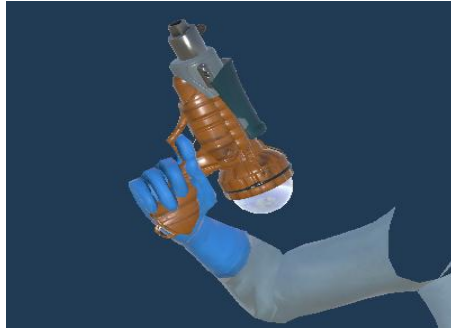


Reload Animation for Plastic Blaster (Same as Glass Blaster and Paper Blaster)

	 <p data-bbox="756 945 1254 981"><i>Reload Animations for Seed Projectile</i></p>
<p data-bbox="347 1012 450 1048">Launch</p>	 <p data-bbox="756 1491 1254 1527"><i>Launch Animations for Seed Projectile</i></p>
<p data-bbox="360 1559 437 1594">Shoot</p>	



Shoot Animation for GermOBlaster



Shoot Animation for Plastic Blaster (Same as Glass Blaster and Paper Blaster)

4.3.2.2 Non-Player Character and Animation Setup

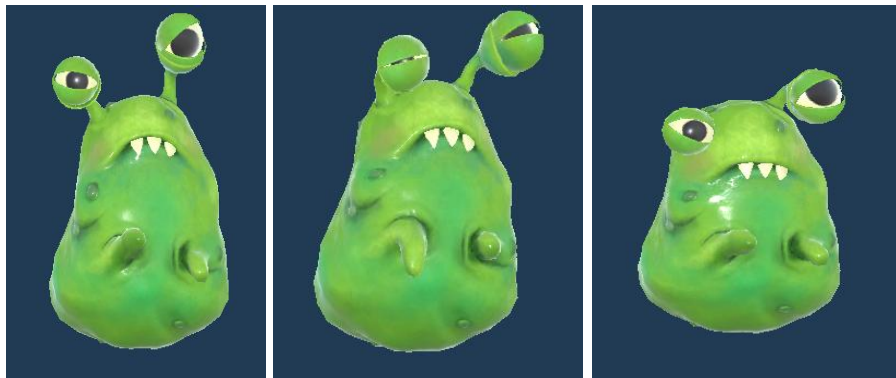
Normal Germ



Normal Germ

Normal Germ is the 3D model downloaded from Unity Asset Store. It is a static germ that contains loot drop objects such as collectable ammo box or health item. Therefore, it only has idle animation. Once the player goes near the germ, health bar will be deducted.

Idle Animation



AI Germ



AI Germ

AI Germ also the 3D model downloaded from Unity Asset Store. It will chase to attack the player if they detect the player's presence. Once the distance between the player and the AI Germ is close, the zombie will attack the player. If the AI Germ attack the player, the player will be deducted life in the health bar. The AI Germ will randomly allocate in different locations in the environment.

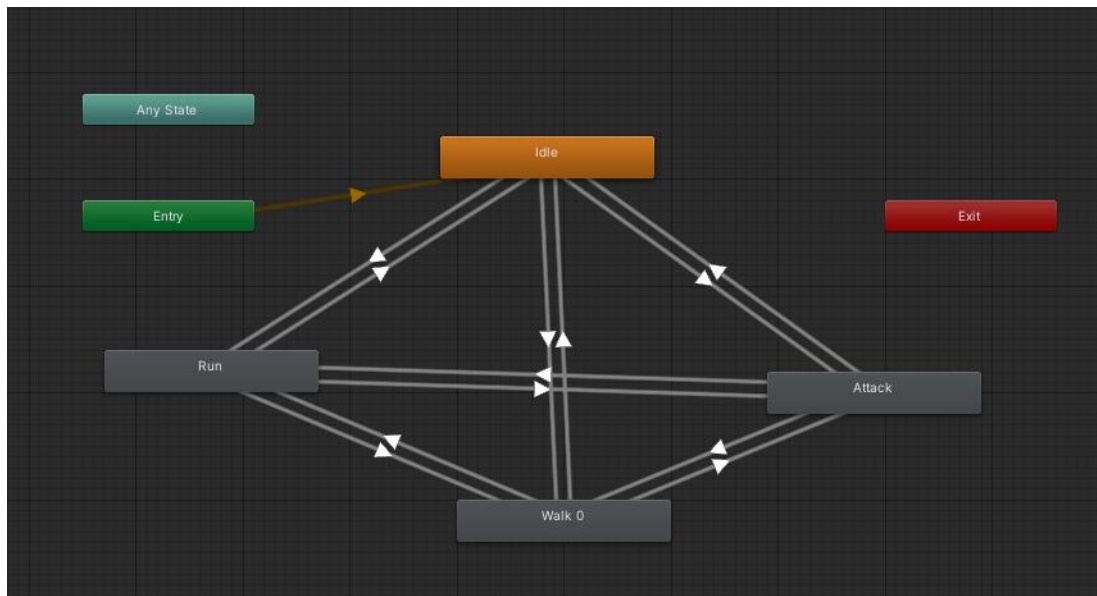


Figure 4.32 Basic Animation - idle, walk, run, attack

Sea Turtles



Sea turtle

It is a 3D model that downloaded from Sketchfab and re-modelling and re-texturing in Blender. Figure 4.33 shows the texture paint of the seaturtle in Blender. It will first needed to unwrap the sea turtle mesh and then only can follow by painting the texture mesh.

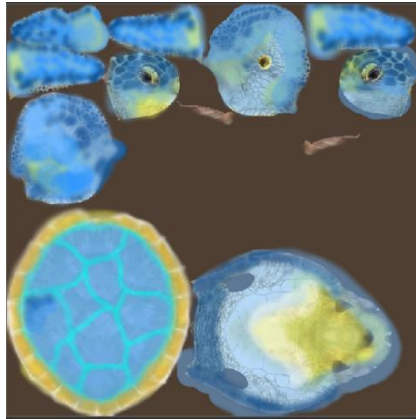


Figure 4.33 Texture Paint in Blender

To make it easier to animate in the Unity, armature bone is added to the 3D model before export to Unity project.

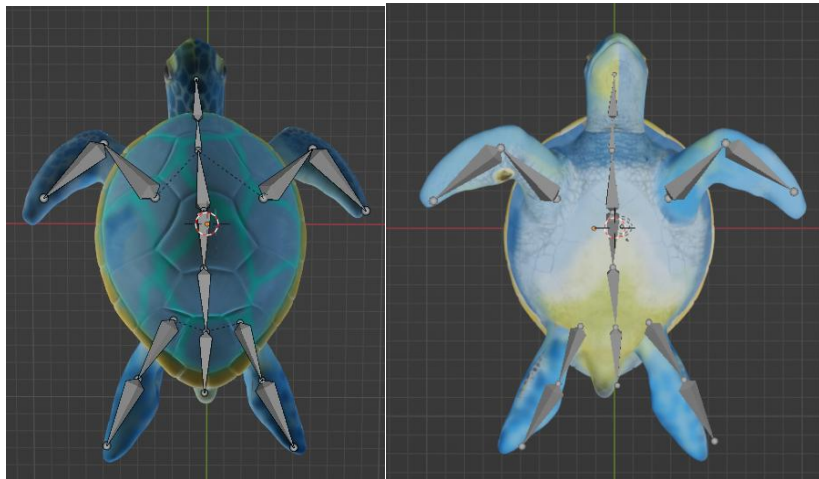


Figure 4.34 Armature bone is added to the Sea turtle 3D model

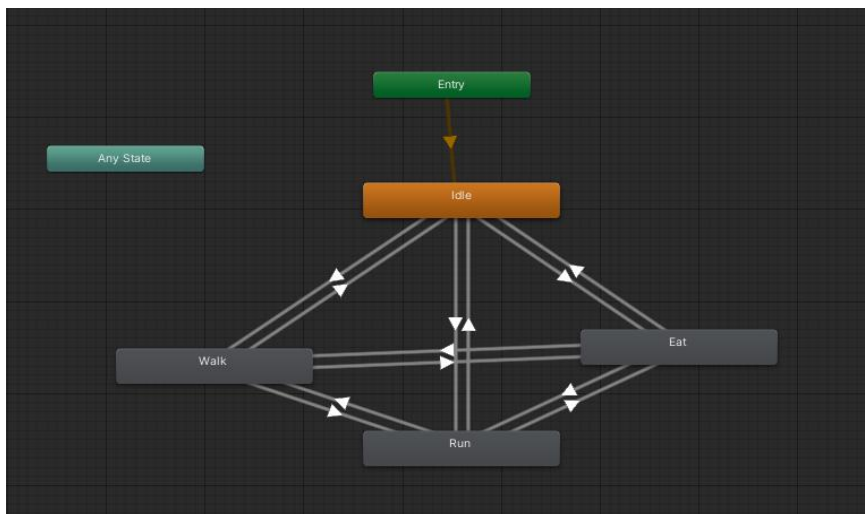
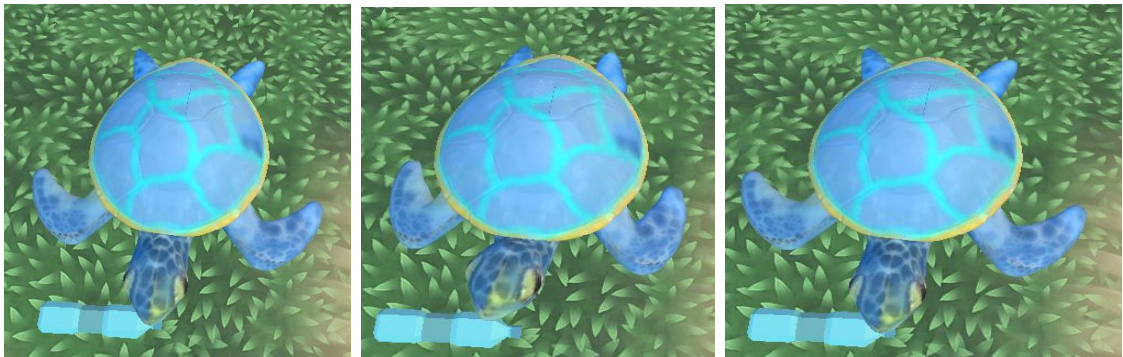


Figure 4.35 Basic Animation - idle, walk, run, eat

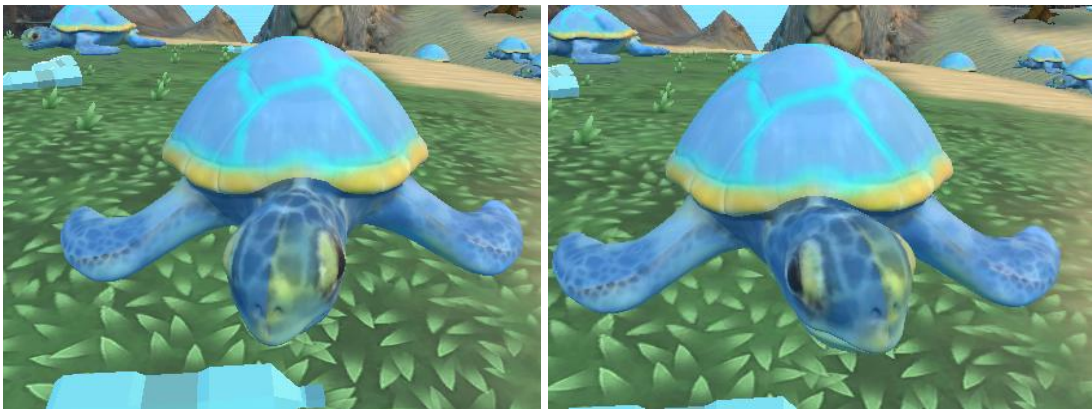
Idle Animation



Walk/Run Animation



Eat Animation



4.3.3 Development of Game Application

The development of the game application in Unity software begins once the multimedia and graphic content has been completed.

4.3.3.1 Implementation of Main Menu Scene

Figure 4.31 shows the main menu interface of the game application. There are four buttons in this scene which are the Tutorial button, the Start button, the How To Play button and the Quit button. When the player clicks the Tutorial button, it will start with an animation to introduce the story of Green Genie. Then, the player will get into a tunnel to start learning all the basic movements and game rules before entering Level 1. If the Start button is clicked, the game will directly start with Level 1 of the game application. Clicking the “How To Play” button will open the game rules for players to view while clicking the “Quit” button will close the application.



Figure 4.31 Main Menu User Interface

Figure 4.32 to Figure 4.37 show the contents of the How To Play menu. Instructions about character movement control, weapon control and the use of each weapon will be provided in this menu. The player can swipe the panels in the How To Play menu left and right to view all the instructions or click the buttons at the bottom of the menu. When the button is pressed, a yellow light will illuminate. When the “X” button at the top right corner is pressed, the menu will be closed and back to the main menu.

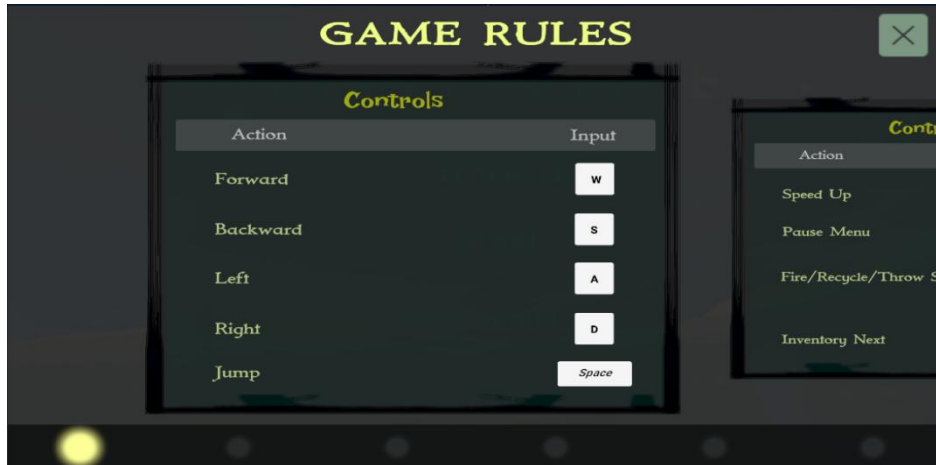


Figure 4.32 First panel in How To Play Interface

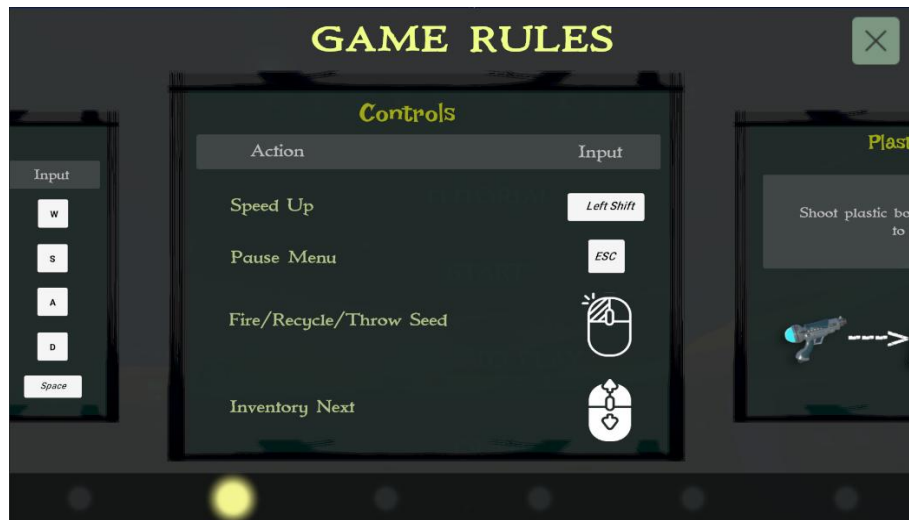


Figure 4.33 Second panel in How To Play Interface



Figure 4.34 Second panel in How To Play Interface



Figure 4.35 Second panel in How To Play Interface



Figure 4.36 Second panel in How To Play Interface



Figure 4.37 Second panel in How To Play Interface

4.3.3.2 Implementation of Introduction Scene: Genie Land

The Introduction Scene starts with an animation to introduce the story of Green Genie. It shows that there is a mysterious Genie Land hiding somewhere in the universe where humans could not see. Jinn, one of the genies, acts as the game's main character. She is a descendant of Genie Land's Godmother.

They live off of the energy released by The Tree of Life (Spirit Tree). Jinn was told one day by small genies that sewage from the human world was slowly flowing into the ancestral river, harming the development of the Spirit Tree. At the same time, Godmother has spent several months in the spirit world. They don't know when she'll return.

As a result, Jinn, as the Godmother's descendent, must take on the responsibility of protecting both the human environment and their Spirit Tree. Despite the fact that Jinn has no understanding of how to manage all of the waste in the human world, he has the ability to get the Godmother's artefacts, which are the Plastic Blaster, Glass Blaster, Paper Blaster, and GermOBlaster. Thor is the Godmother's Retired Valet, who has been following her for generations. He will lead Jinn to a tunnel and teach him the essential techniques of the human world. Jinn will then journey to the human world via the tunnel once he has mastered all of the essential skills. Jinn's mission to fight environmental pollution begins.

Figure 4.38 to Figure 4.48 shows the storyline and dialogue between Jinn and the small genie.

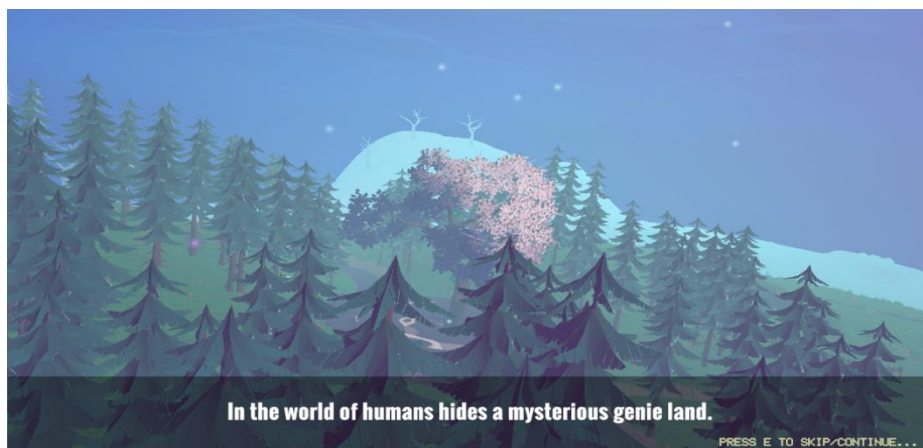


Figure 4.38 Screenshot of the storyline of the Introduction scene.

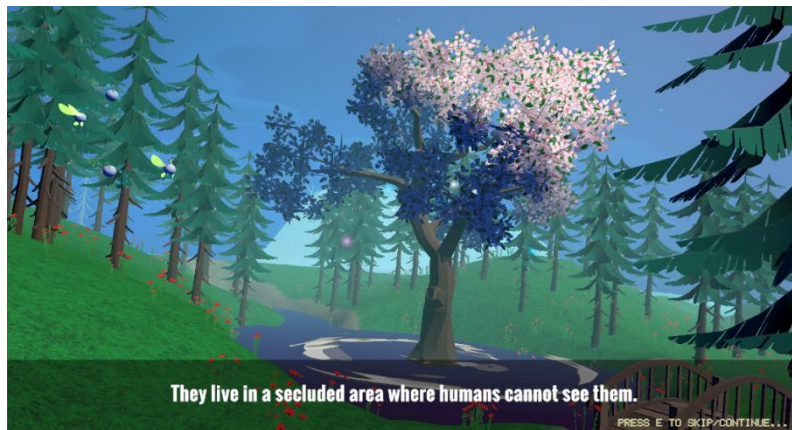


Figure 4.39 Screenshot of the storyline of the Introduction scene.



Figure 4.40 Screenshot of the storyline of the Introduction scene.



Figure 4.41 Dialogue between Jinn and small genie

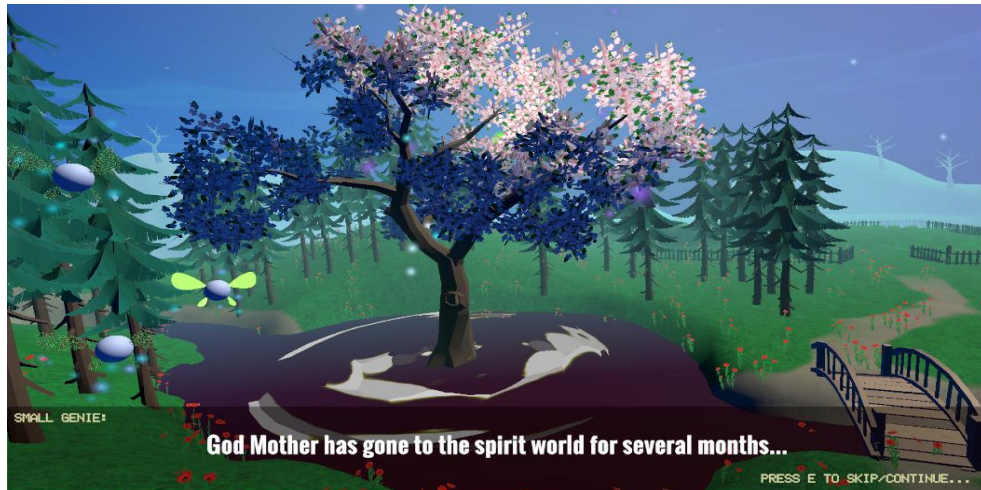


Figure 4.42 Dialogue between Jinn and small genie



Figure 4.43 Dialogue between Jinn and small genie



Figure 4.44 Dialogue between Jinn and small genie



Figure 4.45 Dialogue between Jinn and small genie



Figure 4.46 Dialogue between Jinn and small genie



Figure 4.47 Dialogue between Jinn and small genie



Figure 4.48 Dialogue between Jinn and small genie

There are two cameras placed in this scene. Figure 4.49 shows the animation for the first camera which is moving from the top of the purple boundary to the location of Spirit Tree. After pressing the E button until the conversation between Jinn and Small Genies begins, the scene will switch to the second camera, which will display the river with polluted water in more clarity. Then, the player will enter to Tutorial Scene after the conversation is end.

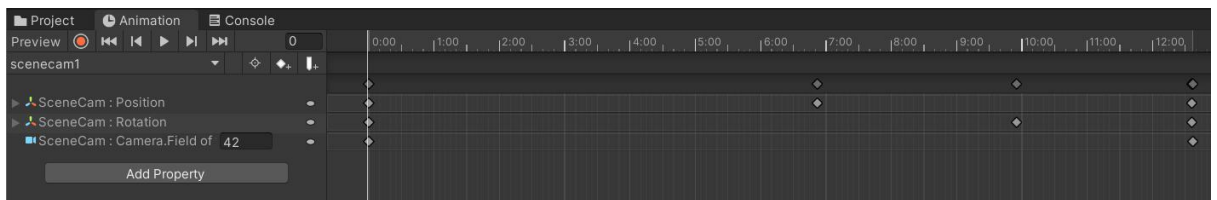


Figure 4.49 Animation for the First Scene Camera

4.3.3.3 Implementation of Tutorial Scene: Tunnel in Genie Land

This scene will guide the player to move using the WASD keys, W is up, A is left, S is down, and D is right while Left Shift Key is to speed up the character movement. The player is also able to move the mouse to look around and left-click the mouse button to shoot. Figure 4.50 shows the first view of the player character in the Tutorial Scene.

Figure 4.51 shows the tutorial interface that guides the player can scroll the mouse wheel or press the number pad to switch weapons. Figure 4.52 depicts the tutorial interface that helps the player through the process of collecting bottles, whereas

Figure 4.53 depicts the tutorial interface that guides the player through the process of shooting the bottles turned into ammo for the Plastic Blaster to the Bin. Figure 4.54 is the tutorial interface that guides the player to shoot the germs and send the red message that states poor waste management could lead to the spread of germs or diseases.

The mission bar at the top right corner contains the tasks that have to be completed before entering Level 1. The countdown time at the top left corner is the time set for the player to complete all the tasks in a limited time frame.

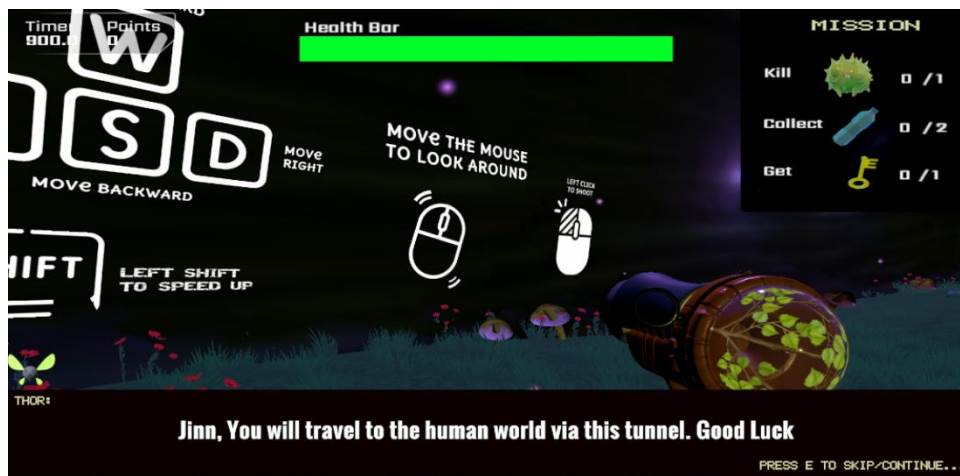


Figure 4.50 First View of the player character in Tutorial Scene.



Figure 4.51 Tutorial to guide the player to switch weapons.



Figure 4.52 Tutorial to guide the player to collect bottles.



Figure 4.53 Tutorial to guide the player to shoot bottles to the recycle bin.



Figure 4.54 Tutorial to guide the player to kill germs.

When the player presses the ESC button, the pause menu will be shown. Figure 4.55 shows the Pause Menu User Interface. The player is able to click the Resume Game button to continue the gameplay and restart the level by clicking the Restart Level button. If the player needs to refer again to the player movement controls or the game rules, he/she can press the Controls/Help button. The interface for Controls/Help is the same as the How to Play Menu in the main menu. (Refer back to Figure 4.32 to Figure 4.37)



Figure 4.55 Pause Menu User Interface

4.3.3.4 Implementation of Level 1 Scene: Trash Island

Game Interfaces for Level 1

Before the player starts to play Level 1, there are some game guidelines or tips for the player. Table 4.6 shows the game rules interfaces for Level 1.

Table 4.6 Game Rules Interfaces for Level 1





Description:

These are the game guidelines or tips interfaces for the player at the beginning of Level 1 start. Based on the interfaces shown, there are three highlights that the player has to know before starting the game level:

- The player has to collect plastic bottles to prevent the sea turtles dead.
- The player has to collect plastic bottles and shoot to the Orange bin to earn points.
- The player has to pay attention to the messages on the wooden board.



Description:

Once the player finish reading all the game guidelines, the timer of Level 1 will start to countdown from 900 seconds. The player has to complete all the mission stated at the right top corner in the limited time set. Or else, the player will need to restart the game when times up.

Level 1 consists of two small Trash Islands:

❖ **Trash Island with Orange Bin and sea turtles.**



Figure 4.56 View of Trash Island with Orange Bin and sea turtles from the opposite Island.

After the player cleared all the plastics bottles to prevent the sea turtle from excessive ingesting the plastic bottles, he/she will need to notice the wooden board that mentioned the player has to shoot the Orange Bin with plastic blaster to earn 200 points to drop a green light and 300 points to drop ammo box.



Figure 4.56 Wooden board with tips in Level 1

The function of green light ball is to guide the player to a location set which will trigger the dialogue box telling the player can jump onto the sea turtles to across the sea. (Refer to Figure 4.57) If the player drop into the sea, an animation showing the player is drown will be shown. (Refer to Figure 4.58)



Figure 4.57 Guidelines or tips interfaces for the player.



Figure 4.58 Sea turtles swim up to help the player jump across the sea.

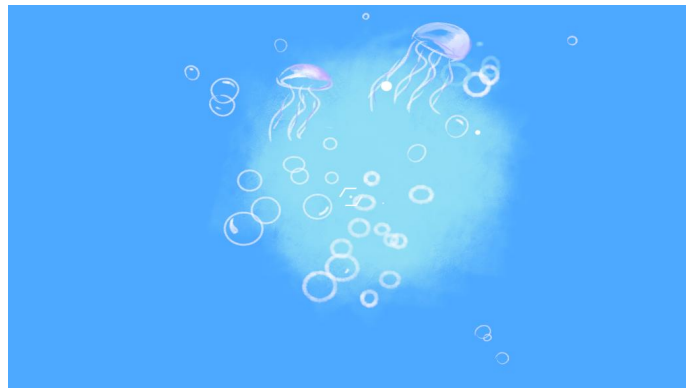


Figure 4.58 Drowning Interface

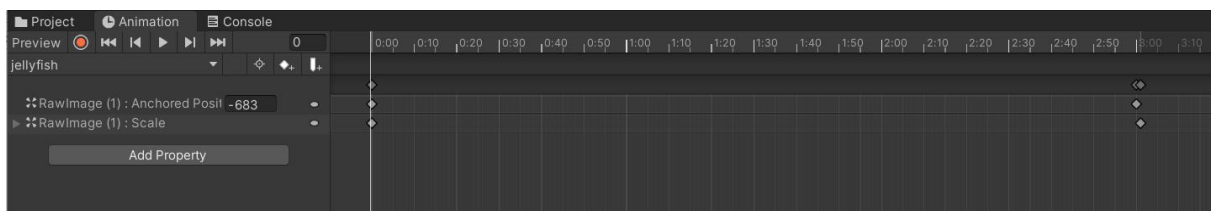


Figure 4.59 Drowning Animation

The player will back to the nearest checkpoint before he/she drop into the sea which means that the player need not to restart the game again. Figure 4.60 shows that the vector point for the empty object with collider will be saved as the nearest checkpoint. If the player drops below 2 in Y-axis, the player will get back to the Vector point that has been saved. The detail scripts will be depicted in the Game Mechanic Setup section. (Topic 4.3.4)

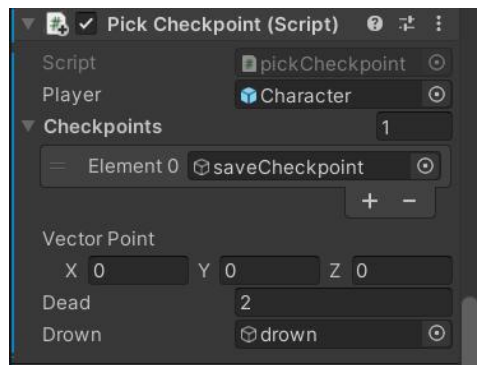


Figure 4.60 Script that assigned to the Character in the Inspector.

Figure 4.61 shows the creating of path with Path Creator Script. The starting point of path is on the top of Orange Bin and the ending point will be at the seaside.



Figure 4.61 Creating the path with Path Creator Script

Figure 4.62 shows the green light ball with particles system effect has the path follower scripts to follow the path set in Figure 4.61.

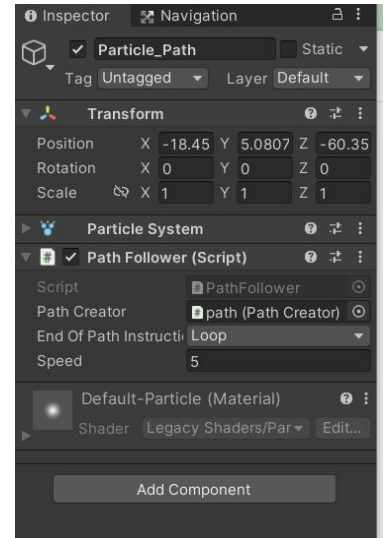


Figure 4.62 Green light ball with path follower script

Figure 4.63 and Figure 4.64 show the results for the green light ball follows the path drawn.

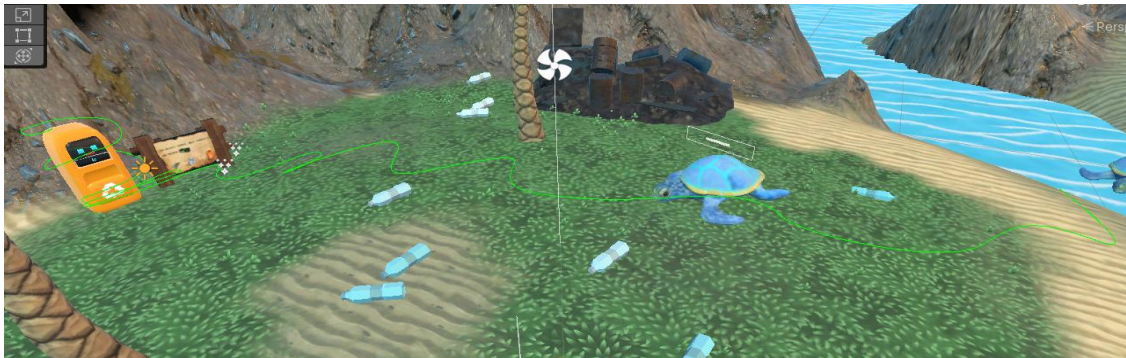


Figure 4.63 The path for the Green light ball to follow.



Figure 4.64 The end result of the Green light ball follows the path in the game scene.

❖ **Trash Island with Germs.**



Figure 4.65 View of Trash Island with Germs from the opposite Island.

When the player reaches the Trash Island with Germs, there are some game guidelines or tips for the player. Table 4.7 shows the game rules interfaces on how to shoot the Germs by using the GermOBlaster.

Table 4.7 Game Rules Interfaces for Shooting the Germ

<p>Oops Jinn, be careful!</p> <p>PRESS E TO SKIP / CONTINUE...</p>	<p>There are the germs that released from the trash!</p> <p>PRESS E TO SKIP / CONTINUE...</p>	<p>Kill them using your Germ Blaster!</p> <p>PRESS E TO SKIP / CONTINUE...</p>
<p>Description:</p> <p>These are the game guidelines or tips interfaces for the player when he/she reaches the Trash Island with Germs. Based on the interfaces shown, there are two highlights that the player has to know:</p> <ul style="list-style-type: none"> ● Germs are released from the trash. ● The player has to switch the weapon to GermOBlaster to clear all the Germs. 		

After the player shooting down a Germ, the Germ will explode and drop either ammo box or key for player to collect.



Figure 4.66 Objects drop from the Germs after getting killed by the player.

Player Rules for Level 1

- The player needs to collect all the plastic bottles and prevent the sea turtles die due to excessive consumption of plastic bottles.
- The player needs to shoot the plastic bottles which has converted to the ammo of Plastic Blaster into the Orange Bin.
- The player needs to switch gun to GermOBlaster to shoot the Germs.
- The player has to kill at least 3 germs, collect 10 plastic bottles and get a key in Level 1. (Figure 4.67)
- If the player is stuck at the middle of the level, then the player needs to restart that level.



Figure 4.67 Mission that need to be completed in Level 1.

Challenges for Level 1

- The player needs to complete all the mission that assigned at the top right corner in 900 seconds.
- The player needs to collect all the plastic bottles as fast as possible to prevent the sea turtles die due to excessive consumption of plastic bottles.
- The player needs to aware of the amount of ammo in the weapon, if the ammo to shoot the Germs is used up but the Germs are still alive, then the player will need to restart the game.

Reward and Punishment for Level 1

- In general, the physical reward of the game has the following:
- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles to the orange bin in Level 1. Each item shot will score 10 points to the corresponding bin.
- Getting ammo box and Portal key from the enemy after the enemy is dead.
- Getting green light balls from the bins if the score of the bins reached 200 points.
- Getting ammo box from the bins if the score of the bins reached 300 points.
- Teleportal will be accessible after the player complete all the mission. (Figure 4.68)
- Advancing to the next level by reaching the end goal.



Figure 4.68 The message interface to notify the player teleportal is accessible.

The more abstract reward in the game that indirectly impacts the gameplay are:

- Improving understanding of improper waste management will result in environmental degradation and the transmission of germs that can cause serious diseases in humans or animals.
- The player learns that plastic bottles are recycled in the Orange Bin.
- Improve the knowledge of the importance of recycling which will save the animals from ingesting plastic and causing death eventually.

Punishments that are in the game have the following:

- When the player cannot complete the tasks assigned in each level, the level fails and has to restart the Level.
- When the player gets attacked by the Germs, the player's health bar will be deducted and the level failed and have to restart when the value of the health bar comes to zero.



Figure 4.69 Game Over Canvas shown when the turtle is dead.

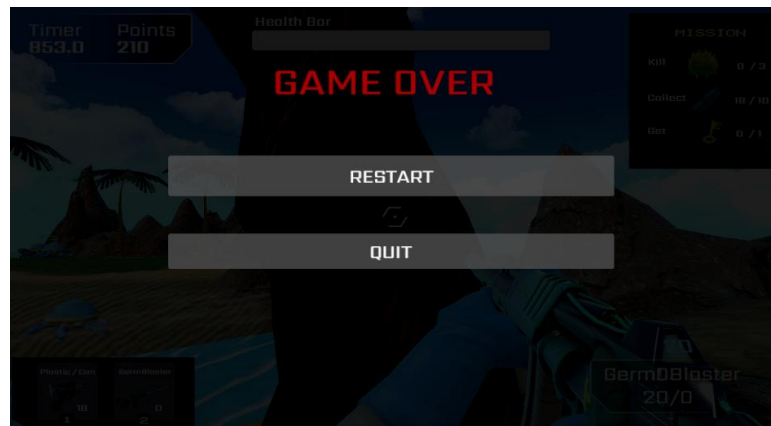


Figure 4.70 Game Over Canvas shown when times up or the player killed by the Germs.

Victory Condition for Level 1

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game and get into the teleportal, the player is able to proceed to the next level.

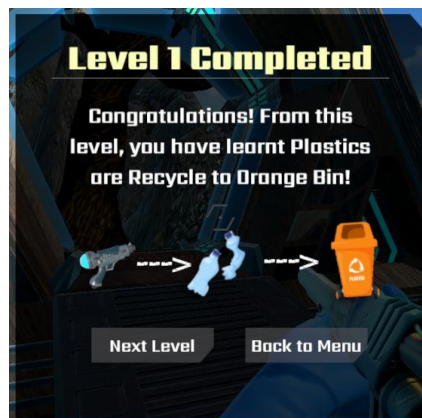


Figure 4.71 Victory Canvas when player complete all the mission and get into the teleportal.

4.3.3.5 Implementation of Level 2 Scene: Trash Fall

Game Interfaces for Level 2

Before the player starts to play Level 2, there are some game guidelines or tips for the player. Table 4.8 shows the game rules interfaces for Level 2.

Table 4.8 Game Rules Interfaces for Level 2

Description:

These are the game guidelines or tips interfaces for the player at the beginning of Level 2 start. Based on the interfaces shown, there are two highlights that the player has to know before starting the game level:

- Germs are released from the trash and they will bring disease.
- The player can plant trees to add health for once.



Figure 4.72 Game View of Level 2

When the player kills the Germs, they will either drop seeds or ammo box for the player. (Refer to Figure 4.73) When the player throw the seed, a tree will be grown. (Refer to Figure 4.74) The seed will heal the player's health bar for 10 points out of 100 points.



Figure 4.73 Seeds and ammo box drop from the Germs after getting killed by the player.

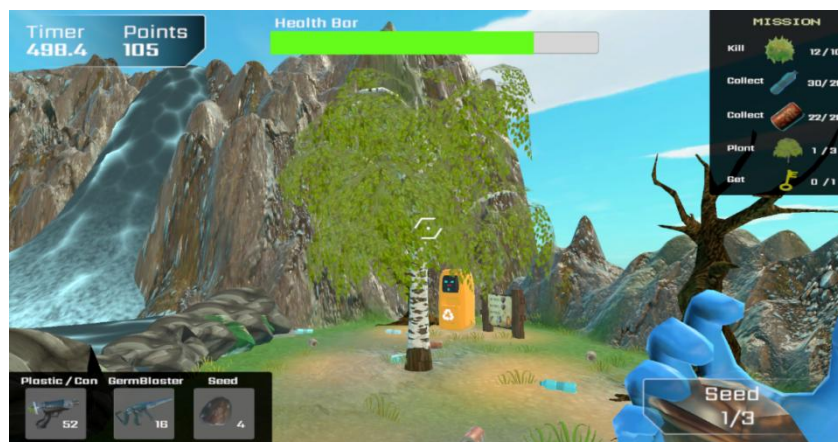


Figure 4.74 Tree planted by the player in Level 2.

Figure 4.75 and Figure 4.76 shows the player has to leap onto the large stone to cross the river.



Figure 4.75 Game View of Level 2



Figure 4.76 To cross the river, the player must leap onto the large stone.

If the player drop into the river accidentally, an animation showing the player is drown wil be shown. (Refer to Figure 4.77)

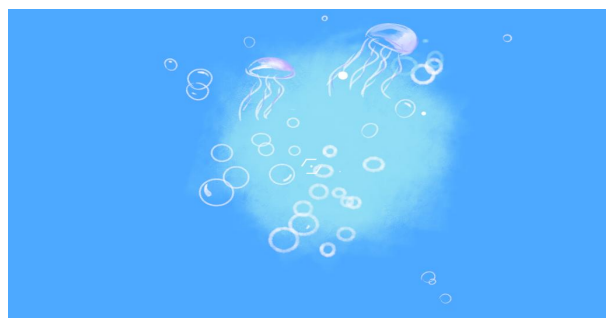


Figure 4.77 Drowning Interface

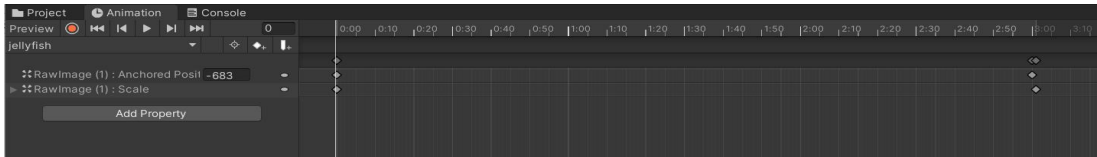


Figure 4.78 Drowning Animation

When the player successfully jumps across the river, he/she has to shoot the plastic bottles and aluminium cans which are converted to ammo for Plastic Blaster to the Orange bin. The player has to shoot the Orange Bin with plastic blaster to earn 100 and 200 points to drop collectable health item and 300 points to drop ammo box. Each ammo box can earn 20 ammo for the GermOBlaster.



Figure 4.79 Wooden board with tips in Level 2

Player Rules for Level 2

- The player needs to collect at least 20 plastic bottles and 20 aluminium cans.
- The player needs to shoot the plastic bottles which has converted to the ammo of Plastic Blaster into the Orange Bin to get Portal key.
- The player needs to switch gun to GermOBlaster to shoot at least 10 Germs.
- The player needs to plant at least 3 trees by switching the weapon to seed projectile.
- If the player is stuck at the middle of the level, then the player needs to restart that level.



Figure 4.80 Mission that needs to be completed in Level 2.

Challenges for Level 2

- The player needs to complete all the mission that assigned at the top right corner in 900 seconds.
- The player needs to aware of the amount of ammo in the weapon, if the ammo to shoot the Germs is used up but the Germs are still alive, then the player will need to restart the game.

Reward and Punishment for Level 2

In general, the physical reward of the game has the following:

- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles and aluminium cans to the orange bin in Level 2. Each item shot will score 10 points to the corresponding bin.
- Getting ammo box and seeds from the enemy after the enemy is dead.
- Getting collectable health from the bins if the score of the bins reached 100 and 200 points.
- Getting Portal key from the bins if the score of the bins reached 300 points.
- Teleportal will be accessible after the player complete all the mission. (Figure 4.81)
- Advancing to the next level by reaching the end goal.



Figure 4.81 The message interface to notify the player teleportal is accessible.

The more abstract reward in the game that indirectly impacts the gameplays are:

- Improving understanding of improper waste management will result in environmental degradation and the transmission of germs that can cause serious diseases in humans or animals.
- The player learns that plastic bottles and aluminium cans are recycled to the Orange Bin.

- Punishments that are in the game have the following:
- When the player cannot complete the tasks assigned in each level, the level fails and has to restart the Level.
- When the player gets attacked by the Germs, the player's health bar will be deducted and the level failed and have to restart when the value of the health bar comes to zero.



Figure 4.82 Game Over Canvas shown when times up or the player killed by the Germs.

Victory Condition for Level 2

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game and get into the teleportal, the player is able to proceed to the next level.



Figure 4.83 Victory Canvas when player complete all the mission and get into the teleportal.

4.3.3.6 Implementation of Level 3 Scene: Dead Tree Island

Game Interfaces for Level 3

Before the player starts to play Level 3, there are some game guidelines or tips for the player. Table 4.9 shows the game rules interfaces for Level 3.

Table 4.9 Game Rules Interfaces for Level 3

 <p>LEVEL 3</p> <p>Oh no.. Ineffective disposal has caused soil contamination..</p>  <p>PRESS E TO SKIP / CONTINUE...</p>	 <p>LEVEL 3</p> <p>The plant cells are damaged and causing the animals lose their habitat..</p>  <p>PRESS E TO SKIP / CONTINUE...</p>
 <p>LEVEL 3</p> <p>Please help the animals to regain their homes!</p>  <p>PRESS E TO SKIP / CONTINUE...</p>	 <p>LEVEL 3</p> <p>SMALL TIPS:</p> <p>There are many glass bottles are thrown. Recycle them to the Chocolate Bin!</p>  <p>PRESS E TO SKIP / CONTINUE...</p>

Description:

These are the game guidelines or tips interfaces for the player at the beginning of Level 3 start. Based on the interfaces shown, there are two highlights that the player has to know before starting the game level:

- Ineffective disposal can cause soil contamination.
- Glass bottles are recycled to Brown Bin.



Figure 4.84 Game View of Level 3

There are no germs in this level but the player will need to learn a new skill which is recycling the glass bottles to the Brown bin. The player also will need to earn seeds to plant at least 5 trees in the game environment. Every 100 points that the player earns from the bins can get a seed and if the player earns 500 point can get a Portal key from each bins.



Figure 4.85 Wooden board with tips in Level 3



Figure 4.86 Wooden board with tips in Level 3



Figure 4.87 Tree planted by the player in Level 3.

Player Rules for Level 3

- The player needs to plant at least 5 trees by switching the weapon to seed projectile.
- The player needs to collect at least 20 plastic bottles and 20 glass bottles.
- The player needs to shoot the plastic bottles which has converted to the ammo of Plastic Blaster into the Orange Bin to get seeds and Portal key.
- The player needs to shoot the glass bottles which has converted to the ammo of Glass Blaster into the Brown Bin to get seeds and Portal key.
- If the player is stuck at the middle of the level, then the player needs to restart that level.



Figure 4.88 Mission that needs to be completed in Level 3.

Challenges for Level 3

- The player needs to complete all the mission that assigned at the top right corner in 900 seconds.
- More recycle bin (Brown Bin) is added into the scene.

Reward and Punishment for Level 3

In general, the physical reward of the game has the following:

- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles ammo to the orange bin and hit the glass bottles ammo to the brown bin in Level 3. Each item shot will score 10 points to the corresponding bin.
- Getting collectable seeds from the bins for every 100 points scored.
- Getting Portal key from the bins if the score of the bins reached 500 points.
- Teleportal will be accessible after the player complete all the mission. (Figure 4.81)
- Advancing to the next level by reaching the end goal.



Figure 4.89 The message interface to notify the player teleportal is accessible.

The more abstract reward in the game that indirectly impacts the gameplays are:

- Ineffective disposal will result in soil contamination and affect the growth of tree.
- The player learns that plastic bottles and aluminium cans are recycled to the Orange Bin while the glass bottles are recycled to the Brown Bin.
- Punishments that are in the game have the following:
- When the player cannot complete the tasks assigned in each level in the limited time set, the level fails and has to restart the Level.

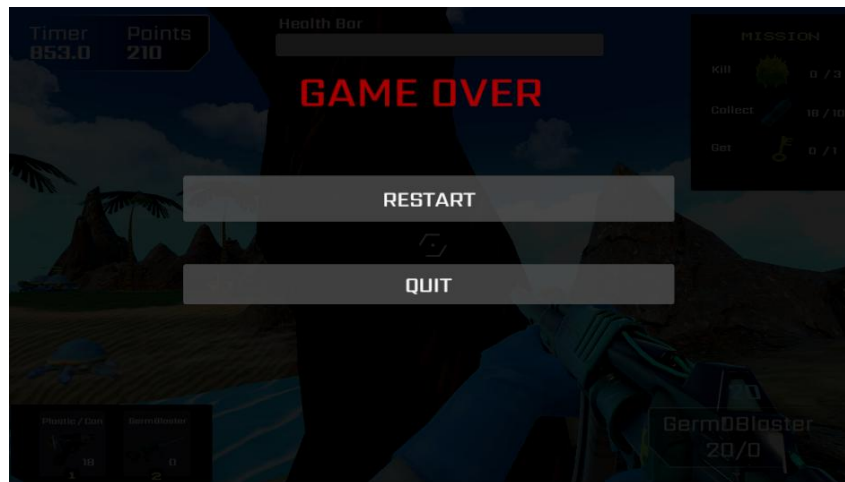


Figure 4.90 Game Over Canvas shown when times up or the player killed by the Germs.

Victory Condition for Level 3

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game and get into the teleportal, the player is able to proceed to the next level.



Figure 4.91 Victory Canvas when player complete all the mission and get into the teleportal.

4.3.3.7 Implementation of Level 4 Scene: Trash City I

Game Interfaces for Level 4

Figure 4.92 shows the game guideline or tip of Level 4 before the player starts to play that level. Based on the interface shown, there is an important message that the player has to know before starting the game level:

- Boxes and cardboard are recycled to Blue Bin.



Figure 4.92 Game Rules Interfaces for Level 4

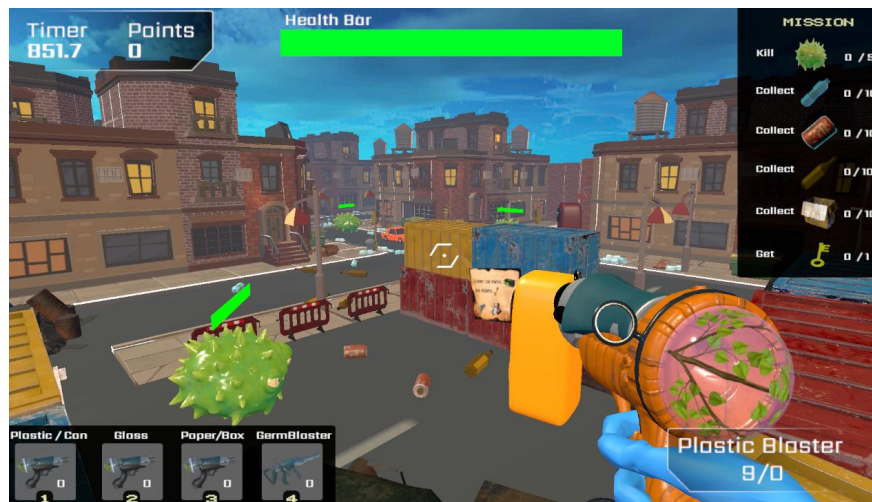


Figure 4.93 Game View of Level 4

Figure 4.94 to Figure 4.96 shows the tips messages that will be pasted near to the bins in this level. Therefore, the player has to observe the game environment. The object that will be dropped from the Germs in Level 4 is the collectable health object.



Figure 4.94 Tips messages of recycling plastic bottles in Level 4



Figure 4.95 Tips messages of recycling boxes/cardboard in Level 4



Figure 4.96 Tips messages of recycling glass bottles in Level 4

Player Rules for Level 4

- The player needs to switch gun to GermOBlaster to shoot at least 10 Germs.

- The player needs to collect at least 10 plastic bottles, 10 aluminium cans, 10 glass bottles and 10 boxes or cardboards.
- The player needs to shoot the plastic bottles which has converted to the ammo of Plastic Blaster into the Orange Bin to get Portal key.
- If the player is stuck at the middle of the level, then the player needs to restart that level.



Figure 4.97 Mission that needs to be completed in Level 4.

Challenges for Level 4

- The player needs to complete all the mission that assigned at the top right corner in 900 seconds.
- The player needs to aware of the amount of ammo in the weapon, if the ammo to shoot the Germs is used up but the Germs are still alive, then the player will need to restart the game.

Reward and Punishment for Level 4

In general, the physical reward of the game has the following:

- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles and aluminium cans to the orange bin, glass bottles to the brown bin and boxes and cardboards to the blue bin.
- Each item shot will score 10 points to the corresponding bin.
- Getting collectable health from the enemy after the enemy is dead. (Figure 4.98)



Figure 4.98 Collectable health drop from the Germs after getting killed by the player.

- Getting Portal key from the Orange Bin if the score of the bin reached 500 points.
- Teleportal will be accessible after the player complete all the mission. (Figure 4.99)
- Advancing to the next level by reaching the end goal.



Figure 4.99 The message interface to notify the player teleportal is accessible.

The more abstract reward in the game that indirectly impacts the gameplay are:

- Improving understanding of improper waste management will result in environmental degradation and the transmission of germs that can cause serious diseases in humans or animals.
- The player learns that plastic bottles are recycled in the Orange Bin.
- The player learns that glass bottles are recycled in the Brown Bin.
- The player learns that paper materials such as boxes and cardboards are recycled in the Blue Bin.

Punishments that are in the game have the following:

- When the player cannot complete the tasks assigned in each level, the level fails and has to restart the Level.
- When the player gets attacked by the Germs, the player's health bar will be deducted and the level failed and have to restart when the value of the health bar comes to zero.

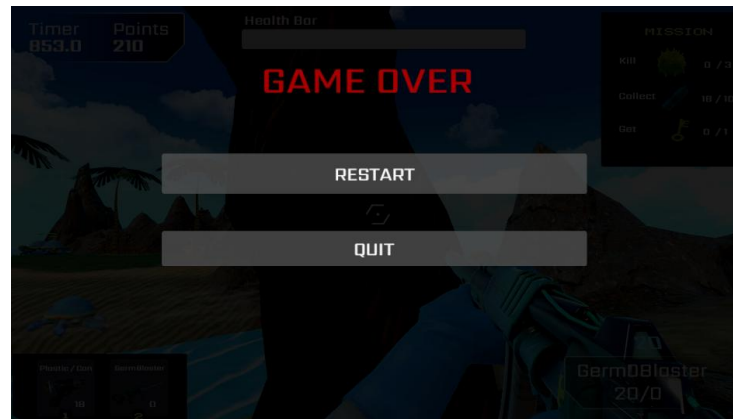


Figure 4.100 Game Over Canvas shown when times up or the player killed by the Germs.

Victory Condition for Level 4

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game and get into the teleportal, the player is able to proceed to the next level.



Figure 4.101 Victory Canvas whe player complete all the mission and get into the teleportal.

4.3.3.8 Implementation of Level 5 Scene: Trash City II

Game Interfaces for Level 5

Figure 4.102 shows the game guideline or tip of Level 5 before the player starts to play that level. Based on the interface shown, there is an important message that the player has to know before starting the game level:

- Every 100 points scored will provide a ammo box.



Figure 4.102 Game Rules Interfaces for Level 5

In this level, there will be no hints message pasted in the environment as Level 4. This is to test whether the player skilled with all the recycling techniques learnt from Level 1 to Level 4. This is level is to recall all the knowledge that has been learnt.



Figure 4.103 Game View of Level 5

Player Rules for Level 5

- The player needs to collect at least 10 plastic bottles, 10 aluminium cans, 10 glass bottles and 10 boxes or cardboard.

- The player needs to shoot the plastic bottles and aluminium cans ammo to the Orange Bin, glass bottles ammo to Brown Bin and dirty boxes/cardboard ammo to the Paper Blaster.
- The player needs to switch gun to GermOBlaster to shoot at least 15 Germs.
- If the player is stuck at the middle of the level, then the player needs to restart that level.



Figure 4.104 Mission that needs to be completed in Level 5.

Challenges for Level 5

- The player needs to complete all the mission that assigned at the top right corner in 900 seconds.
- The player needs to aware of the amount of ammo in the weapon, if the ammo to shoot the Germs is used up but the Germs are still alive, then the player will need to restart the game.

Reward and Punishment for Level 5

In general, the physical reward of the game has the following:

- Getting points after hitting the correct items to the correct bins. For example, the player should hit the plastic bottles and aluminium cans to the orange bin, glass bottles to the brown bin and boxes and cardboard to the blue bin.
- Each item shot will score 10 points to the corresponding bin. Every 100 points earned from each Bins will drop an ammo box.
- Getting collectable health or ammo box from the enemy after the enemy is dead. (Figure 4.105)



Figure 4.105 Collectable health drop from the Germs after getting killed.

- Teleportal will be accessible after the player complete all the mission. (Figure 4.105)
- Advancing to the next level by reaching the end goal.



Figure 4.106 The message interface to notify the player teleportal is accessible.

The more abstract reward in the game that indirectly impacts the gameplay are:

- Improving understanding of improper waste management will result in environmental degradation and the transmission of germs that can cause serious diseases in humans or animals.
- The player learns that plastic bottles are recycled in the Orange Bin.
- The player learns that glass bottles are recycled in the Brown Bin.
- The player learns that paper materials such as boxes and cardboard are recycled in the Blue Bin.

Punishments that are in the game have the following:

- When the player cannot complete the tasks assigned in each level, the level fails and has to restart the Level.

- When the player gets attacked by the Germs, the player's health bar will be deducted and the level failed and have to restart when the value of the health bar comes to zero.

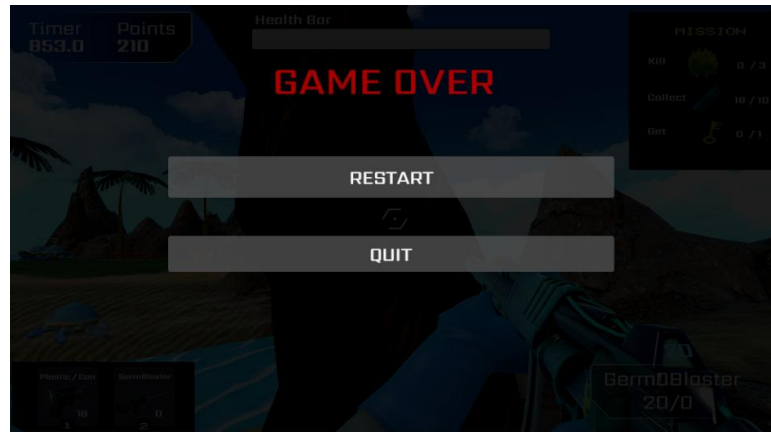


Figure 4.107 Game Over Canvas shown when times up or the player killed by the Germs.

Victory Condition for Level 5

There will be tasks assigned in each level, if the player is able to complete all the tasks in the game and get into the teleportal, the player is able to back to homeland.

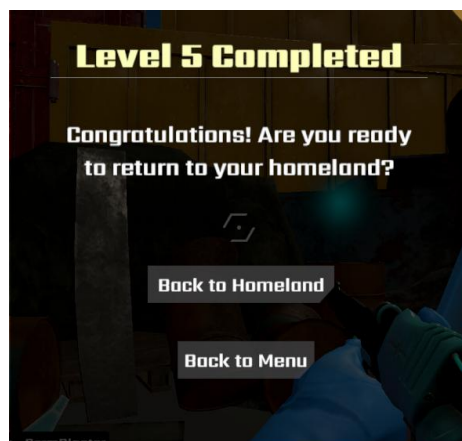


Figure 4.108 Victory Canvas whe player complete all the mission and get into the teleportal.

4.3.3.9 Implementation of End Scene: Back to Genie Land

Game Interfaces for End Scene

Figure 4.109 to 4.113 shows the ending scene interface. The main game character (Jinn) finally complete all his mission and learn all the recycling skills back to her Genie Land.



Figure 4.109 End Scene Interface

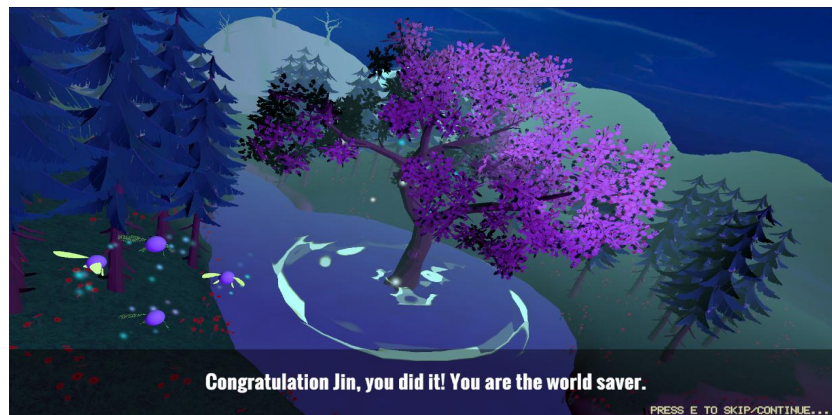


Figure 4.110 End Scene Interface

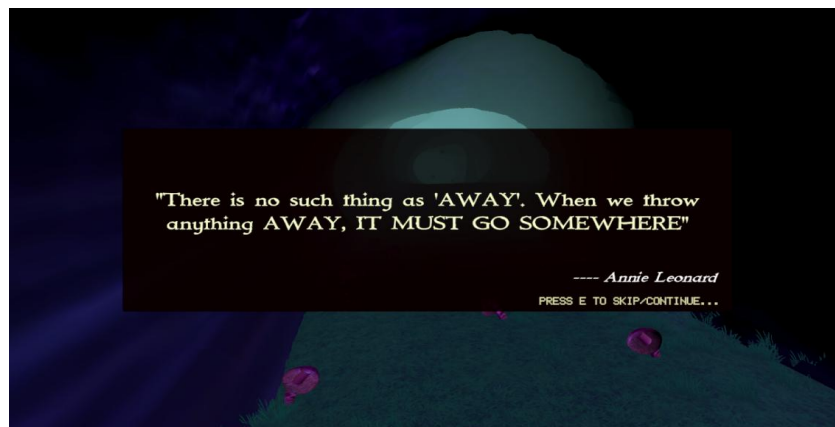


Figure 4.111 End Scene Interface



Figure 4.112 End Scene Interface



Figure 4.113 End Scene Interface

Figure 4.113 shows a panel with 2 selection buttons. If the player choose to press the Play Again button, it will navigate to Level 1 while the player presses Main Menu button, it will navigate to Main Menu of the game application.

4.3.4 Game Mechanic Setup

After all the designing parts are completed, the development of the game application will be started by using Unity 3D software. This section will discuss the implementation of game features and also include all the coding parts for each game feature.

The language used for developing in Unity is the C# programming language. The scripts are written in Microsoft Visual Studio 2022. All the languages that Unity operates with are Object-Oriented Programming (OOP) scripting languages which have syntax, variables, functions and classes. Each script represents a class and may be placed on what Unity calls a "GameObject," which is essentially an object that can contain a large number of scripts that accomplish certain tasks. The figure below shows how to assign multiple scripts to a gameObject.



Figure 4.114 Script Components in a GameObject named "Character"

Now that we comprehend how Unity builds its system, let's dive into the development of the game application.

4.3.4.1 Character Movement and Shooting Control

One of the most fundamental features of the game is how the player controls their characters. This function is developed initially, followed by the others. Players can move their game character with arrow keys or WASD keys to move in 4 directions, left, right, forward, and backward. Besides, players also can press the Spacebar to jump and

the left button of the mouse to shoot the non-player characters (NPC), Germs which will be scattered around the game environment. Besides, the players are also able to shoot the plastic or glass bottles that have been collected to the recycle bin to earn points.

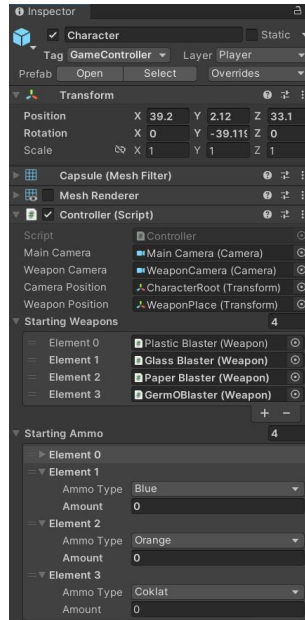


Figure 4.115 Controller component assigned to Character GameObject

In the controller function, the player is also able to collect the bottles and change weapons to shoot.

```
// Move around with WASD
move = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxisRaw("Vertical"));
if (move.sqrMagnitude > 1.0f)
    move.Normalize();

float usedSpeed = m_Grounded ? actualSpeed : m_SpeedAtJump;

move = move * usedSpeed * Time.deltaTime;

move = transform.TransformDirection(move);
m_CharacterController.Move(move);

// Turn player
float turnPlayer = Input.GetAxis("Mouse X") * MouseSensitivity;
m_HorizontalAngle = m_HorizontalAngle + turnPlayer;

if (m_HorizontalAngle > 360) m_HorizontalAngle -= 360.0f;
if (m_HorizontalAngle < 0) m_HorizontalAngle += 360.0f;

Vector3 currentAngles = transform.localEulerAngles;
currentAngles.y = m_HorizontalAngle;
transform.localEulerAngles = currentAngles;
```

```
// Jump (we do it first as
if (m_Grounded && Input.GetButtonDown("Jump"))
{
    m_VerticalSpeed = JumpSpeed;
    m_Grounded = false;
    loosedGrounding = true;
    FootstepPlayer.PlayClip(JumpingAudioClip, 0.8f, 1.1f);
}

bool running = m_Weapons[m_CurrentWeapon].CurrentState == Weapon.WeaponState.Idle && Input.GetButton("Run");
float actualSpeed = running ? RunningSpeed : PlayerSpeed;

if (loosedGrounding)
{
    m_SpeedAtJump = actualSpeed;
}
```

Figure 4.116 Script for Player Movement Control

```

void PickupWeapon(Weapon prefab)
{
    if (m_Weapons.Exists(weapon => weapon.name == prefab.name))
    {
        ChangeAmmo(prefab.ammoType, prefab.clipSize);
    }
    else
    {
        var w = Instantiate(prefab, WeaponPosition, false);
        w.name = prefab.name;
        w.transform.localPosition = Vector3.zero;
        w.transform.localRotation = Quaternion.identity;
        w.gameObject.SetActive(false);

        w.PickedUp(this);

        m_Weapons.Add(w);
    }
}

```

Figure 4.117 Script for picking up weapon

```

void ChangeWeapon(int number)
{
    if (m_CurrentWeapon != -1)
    {
        m_Weapons[m_CurrentWeapon].PutAway();
        m_Weapons[m_CurrentWeapon].gameObject.SetActive(false);
    }

    m_CurrentWeapon = number;

    if (m_CurrentWeapon < 0)
        m_CurrentWeapon = m_Weapons.Count - 1;
    else if (m_CurrentWeapon >= m_Weapons.Count)
        m_CurrentWeapon = 0;

    m_Weapons[m_CurrentWeapon].gameObject.SetActive(true);
    m_Weapons[m_CurrentWeapon].Selected();
}

```

Figure 4.118 Script for changing weapon

```

public int GetAmmo(int ammoType)
{
    int value = 0;
    m_AmmoInventory.TryGetValue(ammoType, out value);

    return value;
}

```

Figure 4.119 Script for collect loot drop objects

```

public void ChangeAmmo(int ammoType, int amount)
{
    if (!m_AmmoInventory.ContainsKey(ammoType))
        m_AmmoInventory[ammoType] = 0;

    var previous = m_AmmoInventory[ammoType];
    m_AmmoInventory[ammoType] = Mathf.Clamp(m_AmmoInventory[ammoType] + amount, 0, 999);

    if (m_Weapons[m_CurrentWeapon].ammoType == ammoType)
    {
        if (previous == 0 && amount > 0)
        {
            m_Weapons[m_CurrentWeapon].Selected();
        }
    }

    WeaponInfoUI.Instance.UpdateAmmoAmount(GetAmmo(ammoType));
}

```

Figure 4.120 Script for change ammo/ collected objects

4.3.4.2 Camera Setting

Since the genre of this game application is first person shooter, the cursor in the screen has to be hidden so that it would not affect the game appearance and game experience. The figures below show the script to hide and show the cursor.

```
void Start()
{
    Cursor.lockState = CursorLockMode.Locked;
    Cursor.visible = false;

    m_IsPaused = false;
    m_Grounded = true;

    MainCamera.transform.SetParent(CameraPosition, false);
    MainCamera.transform.localPosition = Vector3.zero;
    MainCamera.transform.localRotation = Quaternion.identity;
    m_CharacterController = GetComponent<CharacterController>();
}
```

Figure 4.121 Script for locking the cursor

```
public void DisplayCursor(bool display)
{
    m_IsPaused = display;
    Cursor.lockState = display ? CursorLockMode.None : CursorLockMode.Locked;
    Cursor.visible = display;
}
```

Figure 4.122 Script for displaying the cursor while the pause menu shows up

4.3.4.3 NPC AI pathfinding

This feature technology is utilized for the Non-Player Characters (NPCs). There are two types of NPCs in the game application which are the Germs and also the Animals. Germs will attack the player while the animals will move to eat plastic bottles. Therefore, the AI pathfinding technology is to allow the NPCs to follow the waypoints and complete their mission in the game.

There are a total of 2 function scripts needed to make the AI pathfinding technology works which are States.cs and Ai.cs. States.cs is the script that includes the functions to being idle, attack, pursue and patrol while Ai.cs is to assign and set the waypoints route for the NPCs. The combination of these two scripts allows the NPCs to follow the path and complete the tasks at that certain level.

```
// Can the NPC see the player, using a simple Line Of Sight calculation
public bool CanSeePlayer()
{
    Vector3 direction = player.position - npc.transform.position;
    float angle = Vector3.Angle(direction, npc.transform.forward);

    // If player is close enough to the NPC AND within the visible
    if (direction.magnitude < visDist && angle < visAngle)
    {
        return true; // NPC CAN see the player.
    }
    return false; // NPC CANNOT see the player.
}
```

Figure 4.123 Script for checking whether NPC can see the player in State.cs

```

public bool CanAttackPlayer()
{
    Vector3 direction = player.position - npc.transform.position; //
    if (direction.magnitude < shootDist)
    {
        return true; // NPC IS close enough to the player to attack.
    }
    return false; // NPC IS NOT close enough to the player to attack.
}

```

Figure 4.124 Script for checking whether NPC is close to the player in State.cs

```

// Constructor for Idle state.
public class Idle : State
{
    public Idle(GameObject _npc, NavMeshAgent _agent, Animator _anim, Transform _player)
        : base(_npc, _agent, _anim, _player)
    {
        name = STATE.IDLE; // Set name of current state.
    }

    public override void Enter()
    {
        anim.SetTrigger("Idle"); // Sets any current animation state back to Idle.
        base.Enter(); // Sets stage to UPDATE.
    }

    public override void Update()
    {
        if (CanSeePlayer())
        {
            nextState = new Pursue(npc, agent, anim, player);
            stage = EVENT.EXIT; // The next time 'Process' runs, the EXIT stage will run
        }
        // The only place where Update can break out of itself. Set chance of breaking out
        else if (Random.Range(0, 100) < 10)
        {
            nextState = new Patrol(npc, agent, anim, player);
            stage = EVENT.EXIT; // The next time 'Process' runs, the EXIT stage will run
        }
    }

    public override void Exit()
    {
        anim.ResetTrigger("Idle"); // Makes sure that any events queued up for Idle are
        base.Exit();
    }
}

```

Figure 4.125 Script for being idle in State.cs

```

public class Patrol : State
{
    int currentIndex = -1;
    public Patrol(GameObject _npc, NavMeshAgent _agent, Animator _anim, Transform _player)
        : base(_npc, _agent, _anim, _player)
    {
        name = STATE.PATROL; // Set name of current state.
        agent.speed = 2; // How fast your character moves ONLY if it has a path. Not used if
        agent.isStopped = false; // Start and stop agent on current path using this bool.
    }

    public override void Enter()
    {
        float lastDist = Mathf.Infinity; // Store distance between NPC and waypoints.

        // Calculate closest waypoint by looping around each one and calculating the distance
        for (int i = 0; i < npc.GetComponent<Ai>().waypoints.Length; i++)
        {
            Transform thisWP = npc.GetComponent<Ai>().waypoints[i];
            float distance = Vector3.Distance(npc.transform.position, thisWP.position);
            if (distance < lastDist)
            {
                currentIndex = i - 1; // Need to subtract 1 because in Update, we add 1 to
                lastDist = distance;
            }
            waypointCount = i;
        }

        anim.SetTrigger("Walk"); // Start agent walking animation.
        base.Enter();
    }
}

```

Figure 4.126 Script for patrolling in State.cs

```

public class Pursue : State
{
    public Pursue(GameObject _npc, NavMeshAgent _agent, Animator _anim, Transform _player)
        : base(_npc, _agent, _anim, _player)
    {
        name = STATE.PURSUE; // State set to match what NPC is doing.
        agent.speed = 5; // Speed set to make sure NPC appears to be running.
        agent.isStopped = false; // Set bool to determine NPC is moving.
    }

    public override void Enter()
    {
        anim.SetTrigger("Run"); // Set running trigger to change animation.
        base.Enter();
    }

    public override void Update()
    {
        agent.SetDestination(player.position); // Set goal for NPC to reach but navmesh
        if (agent.hasPath) // ...check if agent has a path yet.
        {
            if (CanAttackPlayer())
            {
                nextState = new Attack(npc, agent, anim, player); // If NPC can attack play
                stage = EVENT.EXIT; // Set stage correctly as we are finished with Pursue
            }
            // If NPC can't see the player, switch back to Patrol state.
            else if (!CanSeePlayer())
            {
                nextState = new Patrol(npc, agent, anim, player); // If NPC can't see play
                stage = EVENT.EXIT; // Set stage correctly as we are finished with Pursue
            }
        }
    }
}

```

Figure 4.127 Script for pursuing target in State.cs

```

public class Attack : State
{
    float rotationSpeed = 2.0f; // Set speed that NPC will rotate around to face player.
    //AudioSource shoot; // To store the AudioSource component.
    public Attack(GameObject _npc, NavMeshAgent _agent, Animator _anim, Transform _player)
        : base(_npc, _agent, _anim, _player)
    {
        name = STATE.ATTACK; // Set name to correct state.
        //shoot = _npc.GetComponent<AudioSource>(); // Get AudioSource component for shoot
    }

    public override void Enter()
    {
        anim.SetTrigger("Attack"); // Set shooting trigger to change animation.
        agent.isStopped = true; // Stop NPC so he can shoot.
        //shoot.Play(); // Play shooting sound.
        base.Enter();
    }

    public override void Update()
    {
        // Calculate direction and angle to player.
        Vector3 direction = player.position - npc.transform.position; // Provides the vect
        float angle = Vector3.Angle(direction, npc.transform.forward); // Provide angle of
        direction.y = 0; // Prevent character from tilting.

        // Rotate NPC to always face the player that he's attacking.
        npc.transform.rotation = Quaternion.Slerp(npc.transform.rotation,
            Quaternion.LookRotation(direction),
            Time.deltaTime * rotationSpeed);

        if (!CanAttackPlayer())
        {
            nextState = new Idle(npc, agent, anim, player); // If NPC can't attack player.
            stage = EVENT.EXIT; // Set stage correctly as we are finished with Attack stat
        }
    }
}

```

Figure 4.128 Script for attacking player in State.cs

4.3.4.4 Collecting Recycle Items (Plastic bottles, Glasses, Paper)

To collect the recycling objects, `OnTriggerEnter` is used to cause two `gameObjects` to collide and destroy one of them, resulting in the effect that the object is picked.

```

void Reset()
{
    GetComponent<Collider>().isTrigger = true;
}

void OnTriggerEnter(Collider other)
{
    Controller c = other.GetComponent<Controller>();

    if (c != null)
    {
        c.ChangeAmmo(ammoType, amount);
        Destroy(gameObject);
    }
}

```

Figure 4.129 Script for picking items.

4.3.4.5 Health bar

```

public class Progressbar : MonoBehaviour
{
    public static Progressbar Instance { get; private set; }

    public Slider slider;
    public Gradient gradient;
    public Image fill;

    void OnEnable()
    {
        Instance = this;
    }

    public void SetMaxProgress(int progress)
    {
        slider.maxValue = progress;
        slider.value = 0;

        fill.color = gradient.Evaluate(1f);
    }

    public void SetProgress(int progress)
    {
        slider.value = progress;
        fill.color = gradient.Evaluate(slider.normalizedValue);
    }
}

```

Figure 4.130 Script for filling the progress bar with gradient colour.

The health bar system is the same script as the pollution bar and just changes the tag to enemy since the player will only get hurt by the Germ. Since the Germ gameObject needs to be shot by the player, its parent collider has to be unticked for the IsTrigger function. This causes a problem that the enemy cannot hurt the player using the collider. To solve this, just add an empty object as the child for the Germ, then add the capsule collider to the empty object and tick for the IsTrigger function.

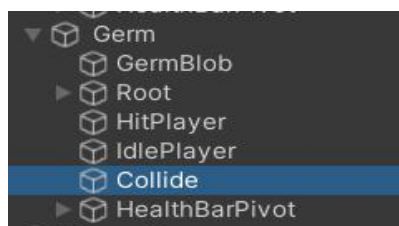


Figure 4.131 Hierarchy of Germ gameObject

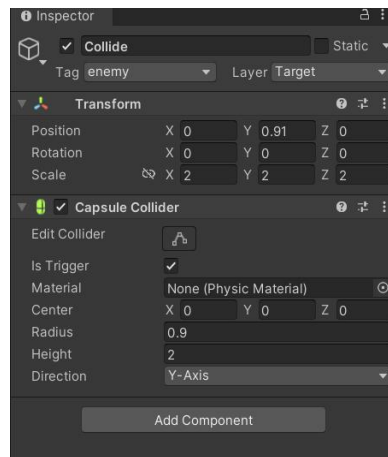


Figure 4.132 Adding Capsule Collider to Germ's parent child - Collide gameObject

```
void Start()
{
    currentProgress = maxProgress;
    Healthbar.Instance.SetMaxProgress(maxProgress);
}

// Update is called once per frame
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.tag == "enemy")
    {
        DeductPoint(20);
        Debug.Log("Point deducted");

        if(currentProgress == 0)
        {
            GameSystem.Instance.StopTimer();
            GameSystem.Instance.GameOver();
        }
    }
}

void DeductPoint(int point)
{
    currentProgress -= point;
    Healthbar.Instance.SetProgress(currentProgress);
}
```

Figure 4.133 Script for health bar

4.3.4.6 Timing and Scoring System

The score will be added to the score panel once the player collected the bottles as ammo and shot to the right colour of the recycle bin. However, if the animals accidentally ingest the plastic bottles and die, the marks will be deducted.

```

if (Physics.Raycast(r, out hit, 1000.0f, ~(1 << 9), QueryTriggerInteraction.Ignore))
{
    Debug.Log(hit.transform.name);

    //if too close, the trail effect would look weird if it arced to hit the wall, so
    if (hit.distance > 5.0f)
        hitPosition = hit.point;

    //this is a target
    if (hit.collider.gameObject.tag == "orange")
    {
        GameSystem.Instance.BinScore(pointValue);
    }
}

```

Figure 4.132 Script for the ammo hitting the collider of bin to add points

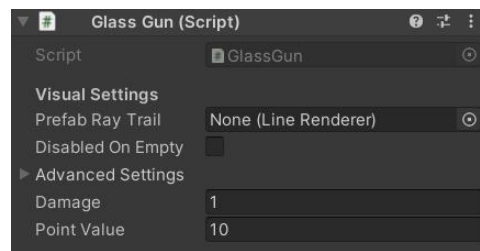


Figure 4.133 The points added can be assigned at point value

```

public void BinScore(int score)
{
    Debug.Log("BIN hit!");

    m_Score += score;

    if (m_Score > 8)
    {
        catMessage.SetActive(false);

        if (m_Score > 20)
        {
            message.SetActive(false);
            message2.SetActive(true);
        }

        if (m_Score > 60)
        {
            message2.SetActive(false);
        }
    }

    GameSystemInfo.Instance.UpdateScore(m_Score);
}

```

Figure 4.134 The scripts for adding score if the ammo hit the correct colour of recycle bin

```

public void CatScore(int score2)
{
    Debug.Log("Cat bite!");

    m_Score -= score2;
    GameSystemInfo.Instance.UpdateScore(m_Score);
}

```

Figure 4.135 The scripts for deducting score if the animal is die

Besides, there is a timing system added to the game application which will affect the score at the end of the level scene. Once the player starts the level, the timer will be started while the timer will be stopped if the player clicks the pause menu. The target (Germs) should be destroyed by the players, if they do not complete the mission, the penalty will be 1 second times the total time that players used to reach the endpoint of the level.

```
void Update()
{
    if (m_TimerRunning)
    {
        m_Timer += Time.deltaTime;

        GameSystemInfo.Instance.UpdateTimer(m_Timer);
    }

    Transform playerTransform = Controller.Instance.transform;

    //UI Update
    MinimapUI.Instance.UpdateForPlayerTransform(playerTransform);

    if(FullscreenMap.Instance.gameObject.activeSelf)
        FullscreenMap.Instance.UpdateForPlayerTransform(playerTransform);
}
```

Figure 4.136 The scripts for deducting score if the animal is die

```
public float GetFinalTime()
{
    int missedTarget = m_TargetCount - m_TargetDestroyed;

    float penalty = missedTarget * TargetMissedPenalty;

    return m_Timer + penalty;
}

public void TargetDestroyed(int score)
{
    m_TargetDestroyed += 1;
    m_Score += score;

    GameSystemInfo.Instance.UpdateScore(m_Score);
}
```

Figure 4.137 The scripts for getting final time

```
public void Display()
{
    gameObject.SetActive(true);

    float time = GameSystem.Instance.RunTime;
    int targetDestroyed = GameSystem.Instance.DestroyedTarget;
    int totalTarget = GameSystem.Instance.TargetCount;
    int missedTarget = totalTarget - targetDestroyed;
    float penaltyAmount = GameSystem.Instance.TargetMissedPenalty * missedTarget;

    TargetDestroyed.text = targetDestroyed + "/" + totalTarget;
    TimeSpent.text = time.ToString("N2") + "s";
    Penalty.text = missedTarget + "*" + GameSystem.Instance.TargetMissedPenalty.ToString("N2")
    + "s = " + penaltyAmount.ToString("N2") + "s";
    FinalTime.text = (time + penaltyAmount).ToString("N2") + "s";

    FinalScore.text = GameSystem.Instance.Score.ToString("N");
}
```

Figure 4.138 The scripts for getting result display

4.3.4.7 Inventory System

An ammo database is created to store all different types of ammo such as plastic bottles, glass bottles and paper balls which will be the projectiles of the gun blaster. To write the ammo database, the Game Database object will be written as a scriptable object so that all the data can be entered more easily. The Figure below shows how easy to add or drop the entries by just editing the Max Id and the Id for each item will be the sequence to change the weapon by just scrolling the middle wheel of the mouse.

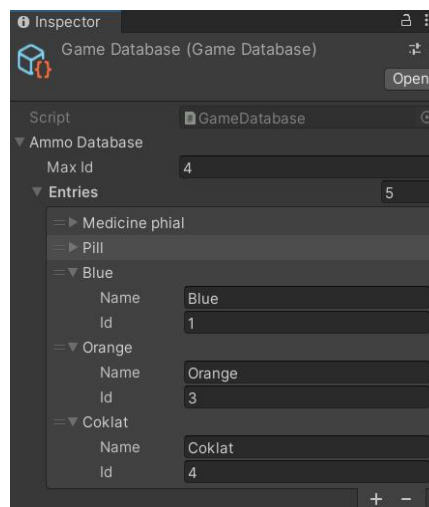


Figure 4.139 The GameDatabase scriptable object

```
[System.Serializable]
public class AmmoDatabase
{
    public int maxId = 0;
    public Queue<int> freeID = new Queue<int>();

    [System.Serializable]
    public class Entry
    {
        public string name;
        public int id;
    }

    public Entry[] entries;
}
```

Figure 4.140 The ammo database scripts for initialize the variables

```
public Entry GetEntry(string name)
{
    return entries.First(entry => entry.name == name);
}

public Entry GetEntry(int id)
{
    return entries.First(entry => entry.id == id);
}
```

Figure 4.141 The ammo database scripts for getting entries

4.3.4.8 Object Loot Drop

When the enemy (Germ) is being killed, a box of ammo will drop out for the player to claim. To create the object loot drop features, there is only one row of scripts needed to add into the scripts in the Enemy gameObject and also make the GameObject public so that the loot drop object can assign to the GameObject.

```
public float health = 5.0f;  
public float m_CurrentHealth;  
public int pointValue;  
public EnemyHealth healthbar;  
public GameObject Ammo;
```

Figure 4.142 Set the Ammo GameObject as public.

```
Instantiate(Ammo, transform.position, Quaternion.identity);
```

Figure 4.143 Script for object loot drop

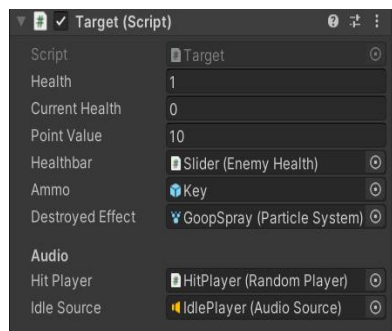


Figure 4.144 Script assigned to Germ GameObject

4.3.4.9 Pause Menu Function

If the pause menu is clicked to display, it will show the pause menu panel and unlock the cursor for the players to choose their selection.

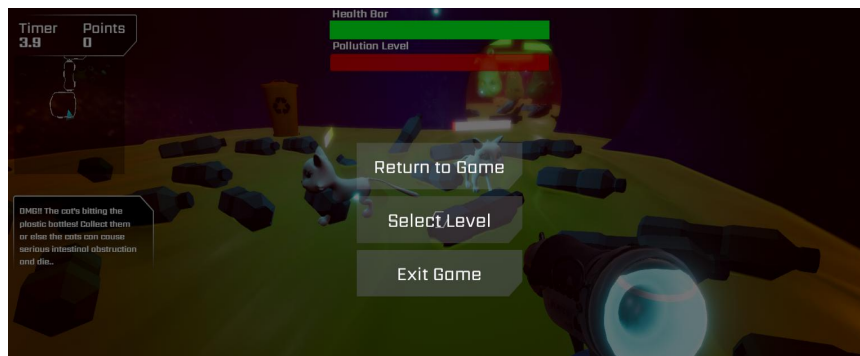


Figure 4.145 Pause Menu of the game

Figure 4.146 below shows the script for the pause menu function.

```
public class PauseMenu : MonoBehaviour
{
    public static PauseMenu Instance { get; private set; }

    void Awake()
    {
        Instance = this;
        gameObject.SetActive(false);
    }

    public void Display()
    {
        gameObject.SetActive(true);
        GameSystem.Instance.StopTimer();
        Controller.Instance.DisplayCursor(true);
    }

    public void OpenEpisode()
    {
        if(LevelSelectionUI.Instance.IsEmpty())
            return;

        UIAudioPlayer.PlayPositive();
        gameObject.SetActive(false);
        LevelSelectionUI.Instance.DisplayEpisode();
    }

    public void ReturnToGame()
    {
        UIAudioPlayer.PlayPositive();
        GameSystem.Instance.StartTimer();
        gameObject.SetActive(false);
        Controller.Instance.DisplayCursor(false);
    }
}
```

```
public void ExitGame()
{
    #if UNITY_EDITOR
        EditorApplication.isPlaying = false;
    #else
        Application.Quit();
    #endif
}
```

Figure 4.146 Scripts for Pause Menu

4.3.4.10 Main Menu Function

```
public class SceneControl : MonoBehaviour
{
    void Start()
    {
        Cursor.visible = true;
        Cursor.lockState = CursorLockMode.None;
    }

    public void Clicked(string nextLevelToLoad)
    {
        SceneManager.LoadScene(nextLevelToLoad);
    }

    public void QuitApp()
    {
        #if UNITY_EDITOR
            EditorApplication.isPlaying = false;
        #else
            Application.Quit();
        #endif
    }
}
```

Figure 4.147 Scripts for Controlling load scene and main menu function.

The figure shows that the cursor will be visible even if the player exits/pauses the game and gets back to the main menu. This is because the playing scene is in the

state of cursor lock, therefore this visibility of the cursor function needs to be added to the main menu scene. There are few function buttons in the main menu which includes start game, tutorial and exit game, so the function scripts will be written as Figure shown above.

4.3.4.11 Save Checkpoints Function

```
private void OnTriggerEnter(Collider other)
{
    if(other.tag == "setpoint")
    {
        Debug.Log("triggerplane");
        vectorPoint = player.transform.position;
    }
}
```

```
private IEnumerator WaitBeforeShow()
{
    yield return new WaitForSeconds(2);
    player.transform.position = vectorPoint;
}
```

Figure 4.148 Scripts for Saving Checkpoints

When the player trigger to the object collider with “setpoint” tag, the system will save the position of the point. If the player drop into the sea or river, the drown animation object will SetActive for 2 seconds. After 2 seconds, the player will get back to the position of the collider.

4.4 Results and Discussion

Following the completion of the GreenGenie game has been developed, the testing is conducted to evaluate the functionality, usability, and efficacy of the game application. The devices such as PCs and laptops can be used to test the game application. The target users for this application will be the elementary students. Therefore, this testing and evaluation will be carried out by identifying primary students, teachers, and parents as evaluators and soliciting their feedback. The alpha testing for Green Genie will use Functional Acceptance Testing (FAT) to test the functionality of project while User Acceptance Test (UAT) will be used for beta testing.



Figure 4.149 A 11-year-old girl named Teh Min Min playing Green Genie game

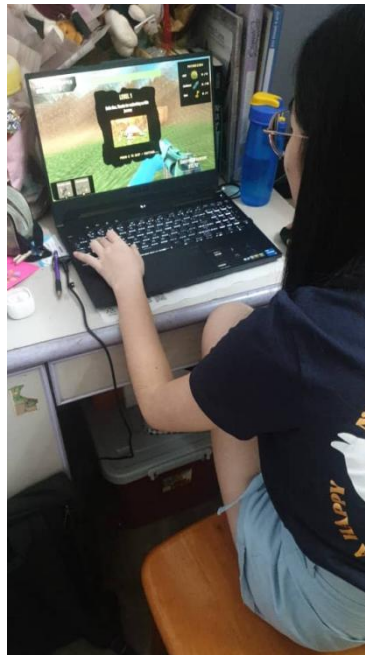


Figure 4.150 A 12-year-old girl named Lee Tze Qian playing Green Genie game



Figure 4.151 A 5-year-old girl named Yeong Min Fei playing Green Genie game



Figure 4.152 A 13-year-old boy named Yeong Zi Qing playing Green Genie game

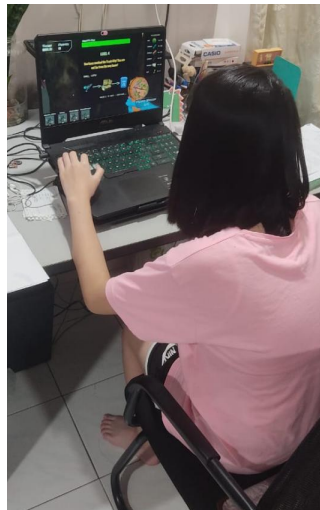


Figure 4.153 A 12-year-old girl named Yeong Pei Ying playing Green Genie game

4.4.1.1 Alpha Testing

Functional Acceptance Test (FAT) is used in Alpha Testing to test and evaluate the functioning of the game application. The evaluators have to test all of the available features in the game application to work and function well. The FAT findings may be found in APPENDIX A. The results of the FAT reveal that all of the functions in the gaming application work properly.

4.4.1.2 Beta Testing

To test the usability and efficacy of the game application, User Acceptance Test (UAT) is used for the beta testing. Green Genie game application is released on Itch.io

for the users to download and play. A Google survey form will be given to the game testers after they have finished playing the game. (Refer to APPENDIX B)

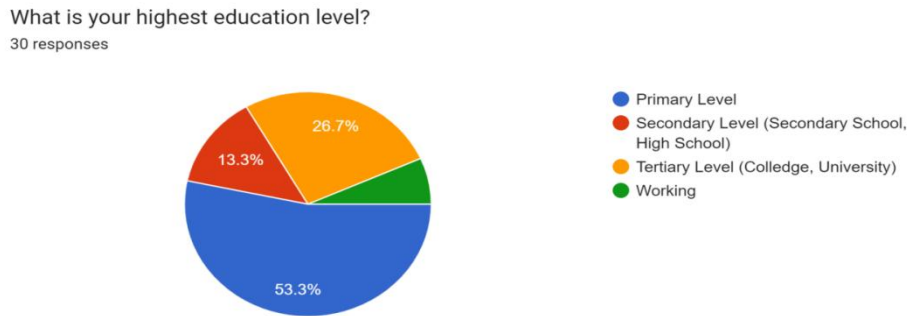


Figure 4.153 Result of “What is your highest education level?”

Figure 4.153 shows that 16 (53.3%) of the 30 total respondents to this survey have a primary level of education. 4 (13.3%) of respondents have a secondary education, 8 (26.7%) have a tertiary degree, and just 2 have actually worked. According to the data, the majority of respondents had a primary education level.

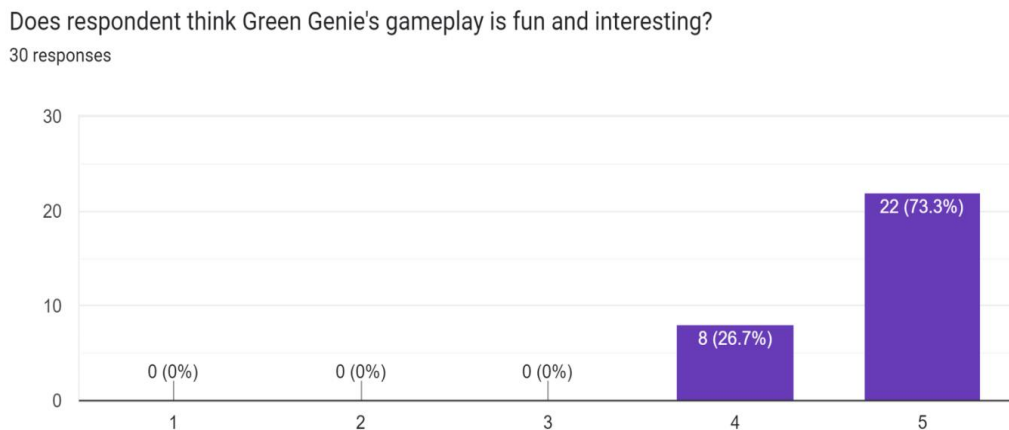


Figure 4.154 Result of “Does respondent think Green Genie's gameplay is fun and interesting?”

Based on Figure 4.154 above, there are 22 out of 30 respondents strongly agree that the gameplay of Green Genie is fun and interesting while the remaining 8 respondents agree with the statement. Therefore, it can be concluded that most of the respondents are satisfied with the gameplay of Green Genie.

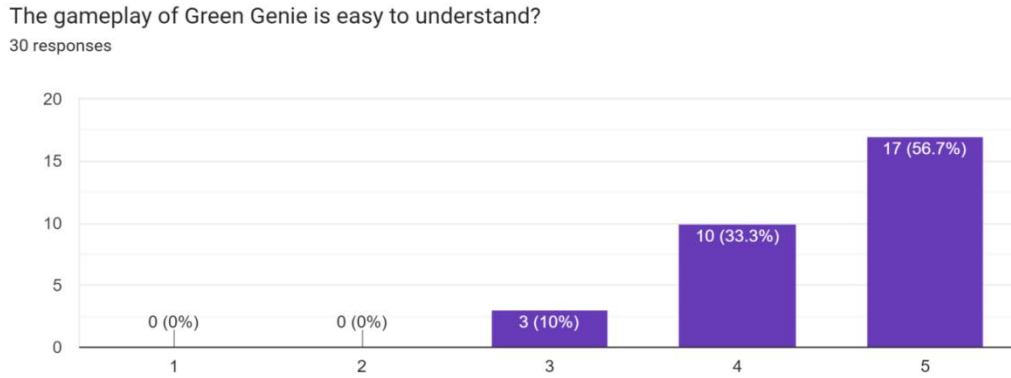


Figure 4.155 Result of “The gameplay of Green Genie is easy to understand? ”

Based on the figure 4.155 above, there are 17 out of 30 respondents strongly agree and 10 out of 30 respondents agree that Green Genie is easy to understand. There are only 3 out of 30 respondents (10%) are neutral with this statement. Therefore, it can be considered that the tutorial in the game has a clear instruction to allow the player to understand the gameplay.

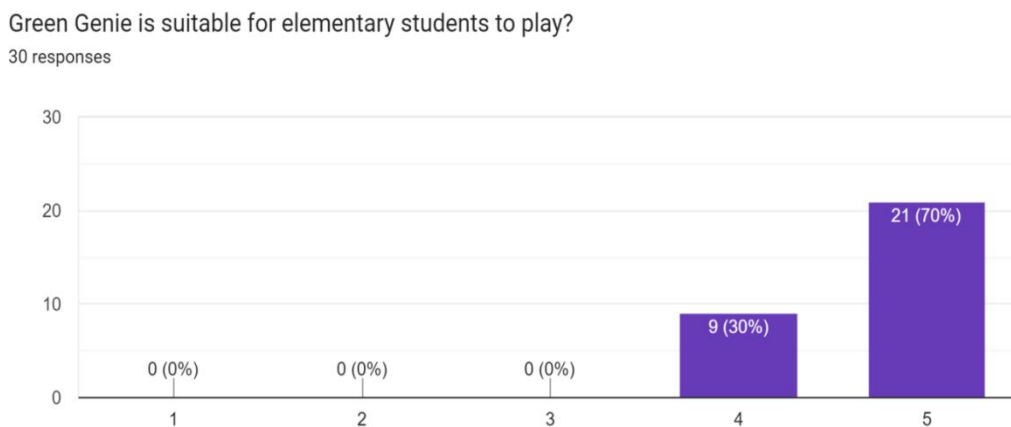


Figure 4.156 Result of “Green Genie is suitable for elementary students to play?”

Based on the figure 4.156 above, the majority of respondents (70%) strongly agree that Green Genie is suitable for elementary students to play and the remaining 30% of respondents are agree with the statement. Therefore, the high percentage of agreement among player indicates that Green Genie is suitable for elementary students to play.

Green Genie has enhanced the recycling knowledge?

30 responses

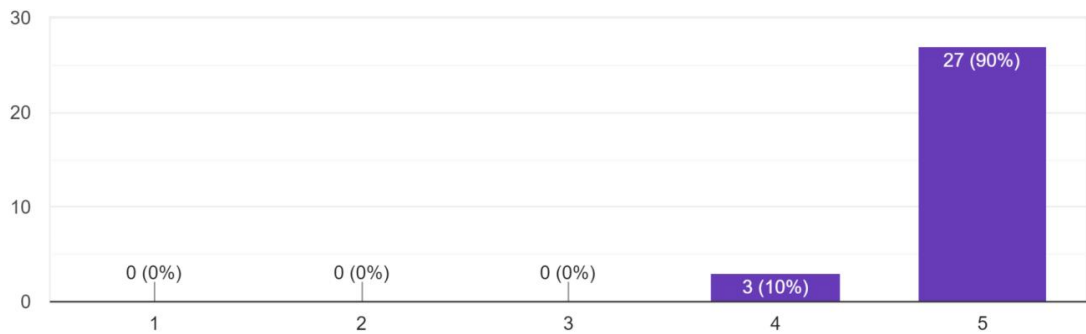


Figure 4.157 Result of “Green Genie has enhanced recycling knowledge?”

Based on the figure 4.157 above, the majority of respondents (90%) strongly agree and 10% of the respondents agree that Green Genie has enhanced the recycling knowledge. As a result, majority of the players agree the Green Genie has enhanced their recycling knowledge.

Green Genie has raised the environmental awareness?

30 responses

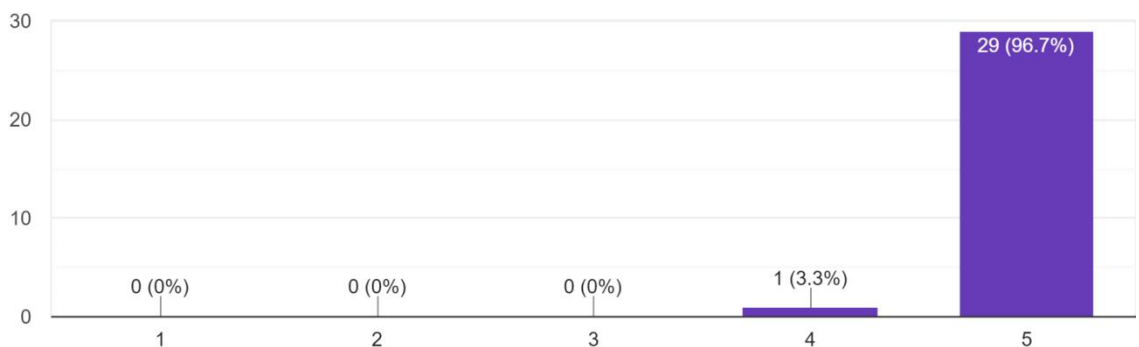


Figure 4.158 Result of “Green Genie has raised the environmental awareness?”

Based on the Figure 4.158 above, the data shows that the majority of the respondents which are 29 out of 30 respondents strongly agree and 1 of the respondent agree with Green Genie has raised the environmental awareness. As a result, it can be concluded that Green Genie game application has successfully delivered the environmental awareness through the gameplay and messages sent in Green Genie.

Which Level of Green Genie do you think is the most difficult to pass?
30 responses

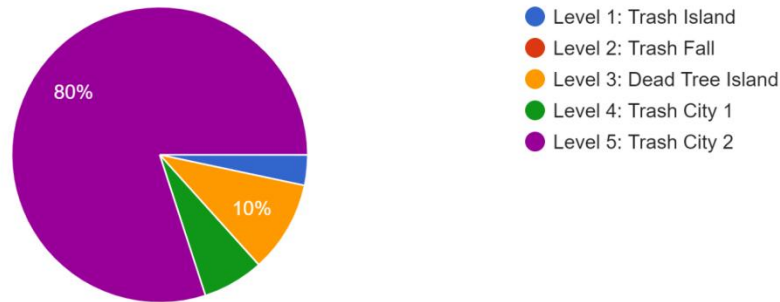
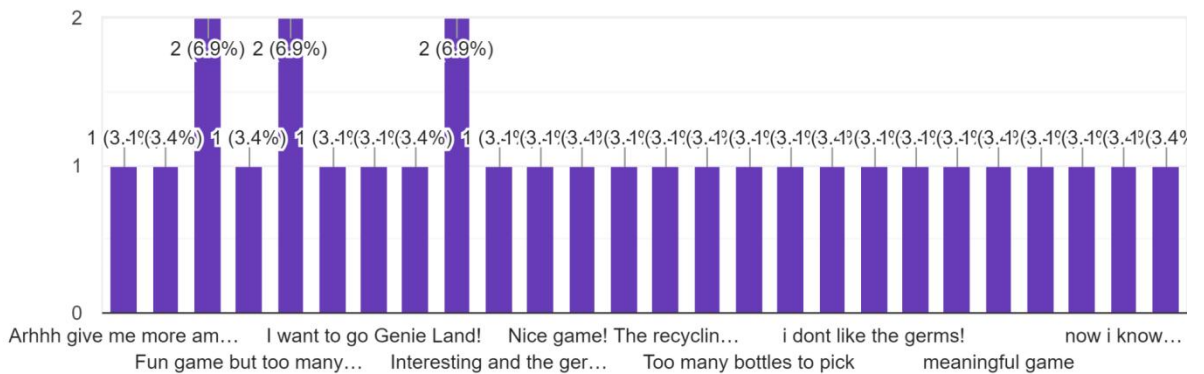


Figure 4.159 Result of “Which Level of Green Genie do you think is the most difficult to pass?”

Based on the Figure 4.159 above, the data shows that the majority of the respondents (24 out of 30 respondents) feel that Level 5 is the most difficult level to pass while only 1 out of 30 respondents feel the Level 1 is difficult to pass. Besides, there are 2 out of 30 respondents and 3 out of 30 respondents feel that Level 4 and Level 3 is difficult to pass respectively.

Kindly leave a short comment(s) on this game.
29 responses



Kindly leave a short comment(s) on this game.	Kindly leave a short comment(s) on this game.
Now I know plastic can be recycle to orange bin!	1
Nice game! The recycling message is very clear to deliver to player!	1
Great Intro!	1
the game is very interesting!	1
nice game	1
Interesting Game	1
Fun Game	2
Interesting Game!	2
Such a Creative game	1
good game!!	1
Fun game but too many trash to collect especially Level 3!!	1
now i know how to recycle!	1
Cut down the number of Germs please...	1
Arhhh give me more ammo box	1
It is fun!	1
Too many bottles to pick	1
Why I always drop into the water???	1
Good	2
Nice game	1
Interesting and the germs are so cute haha	1
meaningful game	1
nice	1
i dont like the germs!	1
i won't simply throw rubbish anymore...	1
interesting	1
I want to go Genie Land!	1

Figure 4.160 Result of “Kindly leave a short comment(s) on this game.”

Figure 4.160 shows the comments from the respondents. Most of the respondents feel that the gameplay of Green Genie is fun and interesting. Based on the comments, the aim of the Gree Genie which is sending the recycling message also has been delivered to the players. There are few suggestion from the young players asking to cut off the amount of bottles and add more ammo box for them.

4.5 Chapter Summary

In conclusion, this chapter has discussed the implementation of Green Genie, the application's testing results, and the application's discussion. The implemented graphic and multimedia contents, character control, game environment design and game mechanics are also explored in depth in terms of how they perform and are implemented. In summary, this chapter has concentrated on the implementation, testing results and discussion of Green Genie, which included the Functional Acceptance Test (FAT) in Alpha Test and User Acceptance Test (UAT) in Beta Test. The project's limits and potential improvements will be further discussed in the next chapter.

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 will discuss the summary of development of Green Genie game application for elementary school children. Based on the UAT testing done in Chapter 4.4, the game application has achieved the objectives of this project mentioned in Chapter 1.3. The first objective in this project is to study the existing environmental sustainability game for elementary school students and this has been done in Chapter 2. Lumino City, Plasticity and Green Space are the existing environmental sustainability games that have been deeply elaborated based on their game features, techniques, strengths and weaknesses in Chapter 2.

The second and also one of the essential objectives to develop this project is to develop an environmental sustainability game that provides recycling knowledge for elementary school students. By achieving this objective, the recycling messages have been injected into every level of Green Genie. For example, the players will get to know plastic bottles are recycled to Orange bin in Level 1. Besides, players also get to know disposal of trash especially plastic bottles will threaten the aquatic life such as the sea turtles. In addition, poor waste management also will cause the spread of germs or virus which will threaten living beings' health too.

Furthermore, the players also get to know aluminium cans also can be recycled to Orange Bin in Level 2, glass bottles can be recycled to Brown Bin in Level 3, boxes and cardboard can be recycled to Blue Bin in Level 4. Level 5 will not give any hints to the player anymore, therefore they have to recall back what they have played and learnt in the previous levels. This is to enhance their memory in recycling.

To evaluate the functionality of the developed environmental sustainability game for elementary school students also one of the objectives in developing this project. The functionality of the developed Green Genie game application has been tested using the alpha testing. (Functional Acceptance Test) All the function for every module and buttons are tested and able to function well.

In short, Green Genie has used an indirect way to send messages to the players about the waste pollution that can bring infections due to the spread of harmful bacteria or germs in the surroundings and also threatens the animals' life, especially aquatic life. There will be recycling awareness, and understanding the need of recycling can assist to safeguard marine life from swallowing too many plastic bottles in every Level of Green Genie.

Since Green Genie features multimedia elements such as animations, audios, and graphics, this gaming application can be used to stimulate and motivate the elementary students to protect the sustainability of environment by developing interest in recycling. Children can learn in an enjoyable and interesting way.

5.2 Limitation and Constraint

During the development phase of the game application, a little issue with the lightning and animation issues surfaced. When the game application is exported out, the lightning of the game application will be darken. To solve this issue, I have adjusted the lighting setting in Unity by increasing the intensity multiplier and add on one more directional light to the game environment. (Window > Rendering > Lighting) Besides, the animation of of the animal, Sea turtles which imported from Blender to Unity also meet some issues. The animation of the 3D models imported to Unity will different from the Blender. Therefore, the animation of the models will eventually created using Unity after the armature of the 3D Model is created using Blender.

In an addition, the short time frame also one of the constraint during the development of game application. There are only five levels implemented for Green Genie due to the limited time.

5.3 Future Work

There are several features that can be added or improve for the future of Green Genie game application for elementary student.

- Developer can add more levels and more items to recycle
- Developer can implement Green Genie in more platforms such as Steam and Google Play Store.
- Developer can add multi-language into the game application (For example: Bahasa Melayu and Chinese Language)
- Developer can add the save and load system so that the players are able to continue and save their progress of game.

REFERENCES

1. Austin Downs and Richard Acevedo. *HOW OUR TRASH IMPACTS THE ENVIRONMENT*. <https://www.earthday.org/how-our-trash-impacts-the-environment/>
2. Eyal Elazar. *User Acceptance Testing (UAT) Process*. <https://www.panaya.com/blog/testing/what-is-uat-testing/>
3. Innocent A. Jereme, I. A., Siwar, C. & Alam, M. M. (2015). *Waste recycling in Malaysia: Transition from developing to a developed country*. Indian Journal of Education and Information Management.
4. *Not-In-My-Back-Yard Syndrome (NIBM)*. (2022) New Sunday Times.
5. Paul Hawken. (Apr 01, 2008). *Blessed Unrest: How The Largest Social Movement In History Is Restoring Grace, Justice, And Beauty To The World*.
6. Project of the Institute for the Study of Human Knowledge. *The Human Journey*. https://humanjourney.us/our-plastic-earth/?gclid=Cj0KCQjw4uaUBhC8ARIsANUuDjXOleA7D0Bkfd5ItkGLuXziVPKkrfFo2gV1rizRcGTHMBiiHRiEkwaApPKEALw_wcB
7. Rafia Afroz, Muhammad Mehedi Masud, Rulia Akhtar, & Jarita Bt Duasa (Aug 01, 2013). *Survey and analysis of public knowledge, awareness and willingness to pay in Kuala Lumpur, Malaysia—a case study on household WEEE management*. Journal of Cleaner Production, 52, 185-193.
8. Seneo Mwamba and Khanyi Mlaba (2021). *12 Plastic Pollution Facts That Show Why We Need To Do More*. https://www.globalcitizen.org/en/content/effects-of-plastic-pollution-facts-you-should-know/?gclid=Cj0KCQjw4uaUBhC8ARIsANUuDjVl1nevO8acOi8hR8uie4a8VjdgPyUhZxUqwX5mYF6llAdO_Sd5IMoaA17_EALw_wcB
9. Scott Stiner. (2016). *Rapid Application Development (RAD): A Smart, Quick And Valuable Process For Software Developers*. <https://www.forbes.com/sites/forbestechcouncil/2016/08/24/rapid-application->

[development-rad-a-smart-quick-and-valuable-process-for-software-developers/?sh=6fc56efe19e8](#)

10. United States Environmental Protection Agency. *Impacts of mismanaged Trash*.
https://19january2021snapshot.epa.gov/trash-free-waters/impacts-mismanaged-trash_.html

APPENDIX A
FUNCTIONAL ACCEPTANCE TEST (FAT)

Tester Name: Goh Mei Kei				
Test Case 1: Main Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Start the game from the Intro Scene and Tutorial level.	Click on the tutorial button	The intro scene and tutorial of the game will be loaded.	/
2	Start the game from the first level.	Click on the start button	The first level of the game will be loaded.	/
3	Go to the game instruction menu	Click on the How To Play button.	The game instruction menu will be popped out.	/
4	Quit the game application	Click on the quit button	The game will be closed.	/

Test Case 2: How To Play Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Show the player the instruction for gameplay	Click on the How To Play button.	The game instruction menu can be viewed.	/
2	Go back to the main menu.	Click on the back button.	The main menu will show up back.	/

Test Case 3: Gameplay Tutorial Level				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Player can skip the dialogue	Press E button	The character is able to skip the dialogue for Intro Scene and conversation between Green	/

			Genie and Jinn.	
2	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
3	Player character can jump	Press the spacebar on the keypad.	The character can jump.	/
4	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
5	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
7	Germs in the gameplay can drop ammo box.	The player character shoots down the germ.	Germs in the gameplay drop ammo box.	/
8	Points can be added.	The Player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	/
9.	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
10	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
11	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 4: Gameplay for Level 1

No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
2	Player character can jump	Press the spacebar on the keypad.	The character can jump.	/
3	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
4	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
5	Animals in the gameplay can eat plastics and die after the health bar is reduced to zero.	Animals will auto-move in their path that has been set in the system when the level starts.	Animals in the gameplay can eat plastics and die after the health bar is reduced to zero.	/
6	The Germs can attack the player character.	The player character goes near the Germ.	The Germs attack the player character.	/
7	Germs in the gameplay can drop ammo box or key.	The player character shoots down the germ.	Germs in the gameplay drop ammo box or key.	/
8	Points can be added.	The player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	/
9	Points can be added to Orange Bin	The player character shoots the plastic bottles ammo to the Orange bin	10 points added for 1 bottles ammo shot to the Orange Bin.	/

10	Green light ball drops from the Orange bin	The player character shoots the plastic bottles ammo to the Orange bin and earn 200 points.	Green light ball drop from the Orange bin.	/
11	Ammo box drops from the Orange bin	The player character shoots the plastic bottles ammo to the Orange bin and earn 300 points.	Ammo box drops from the Orange bin.	/
12	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
13	Drown scene appears for 2 seconds	Once the player drop into the sea	Drown scene appears for 2 seconds	/
14	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
15	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 5: Gameplay for Level 2				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
2	Player character can jump	Press the space bar on the keypad.	The character can jump.	/
3	Player character can speed up the movement	Press W/A/S/D together with Left Shif key	Player character can speed up the movement	/

4	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
5	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
6	The Germs can attack the player character.	The player character goes near the Germ.	The Germs attack the player character.	/
7	Germs in the gameplay can drop ammo box	The player character shoots down the germ.	Germs in the gameplay drop ammo box.	/
8	Points can be added.	The player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	/
9	Points can be added to Orange Bin	The player character shoots the plastic bottles ammo or aluminium cans to the Orange bin	10 points added for 1 bottles ammo shot to the Orange Bin.	/
10	Collectable health item drops from the Orange bin	The player character shoots the plastic bottles ammo to the Orange bin and earn 100 and 200 points.	Collectable health item drops from the Orange bin.	/
11	Collectable Portal key drops from the Orange bin	The player character shoots the plastic bottles ammo to the Orange bin and earn 300 points.	Collectable Portal key drops from the Orange bin.	/

12	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
13	Drown scene appears for 2 seconds	Once the player drop into the sea	Drown scene appears for 2 seconds	/
14	Each seeds can heal the player once with 10% of health bar	Once the player go near the tree planted by the seed	Each seeds can heal the player once with 10% of health bar	/
15	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
16	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 6: Gameplay for Level 3				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
2	Player character can jump	Press the spacebar on the keypad.	The character can jump.	/
3	Player character can speed up the movement	Press W/A/S/D together with Left Shift key	Player character can speed up the movement	/
4	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
5	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
6	Points can be added.	The player character shoots the items to the	Points added.	/

		bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.		
7	Points can be added to Orange Bin and Brown Bin	The player character shoots the plastic bottles ammo or aluminium cans to the Orange bin and glass bottles ammo to Brown Bin	10 points added for 1 bottles ammo shot to the Orange Bin and Brown Bin.	/
8	Collectable seeds and Portal Key drop from the Orange bin and Brown Bin	The player character shoots the plastic bottles ammo to the Orange bin and earn points. Every 100 points drop a seed and 500 points drop Portal Key.	Collectable seeds and Portal Key drop from the Orange bin.	/
9	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
10	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
11	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 7: Gameplay for Level 4				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)

1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
2	Player character can jump	Press the spacebar on the keypad.	The character can jump.	/
3	Player character can speed up the movement	Press W/A/S/D together with Left Shif key	Player character can speed up the movement	/
4	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
5	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
6	The Germs can attack the player character.	The player character goes near the Germ.	The Germs attack the player character.	/
7	Germs in the gameplay can drop collectable health item	The player character shoots down the germ.	Germs in the gameplay drop ammo box.	/
8	Points can be added.	The player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	/
9	Points can be added to Orange Bin	The player character shoots the plastic bottles ammo or aluminium cans to the Orange bin	10 points added for 1 bottles ammo shot to the Orange Bin	/
10	Points can be added to	The player	10 points added for	/

	Brown Bin	character shoots the glass bottles ammo to Brown Bin	1 bottles ammo shot to the Brown Bin.	
11	Points can be added to Blue Bin	The player character shoots the dirty boxes and cardboard ammo to Blue Bin	10 points added for 1 dirty boxes and cardboard ammo shot to the Blue Bin.	/
12	Collectable ammo box drops from the Orange bin, Blue Bin and Brown Bin.	The player character shoots the plastic bottles, glass bottles or boxes of ammo to the Orange bin, Brown Bin and Blue Bin to earn points respectively. Every 100 points drop a ammo box for all Bins.	Collectable ammo box drops from the Orange bin, Blue Bin and Brown Bin.	/
13	Collectable Portal Key drops from the Orange bin.	The player character shoots the plastic bottles of ammo to the Orange bin to earn points. 500 points drop a Portal key.	Collectable Portal Key drops from the Orange bin.	/
14	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
15	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
16	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 8: Gameplay for Level 5				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)

1	Player character can move forward, backward, left and right.	Press the W/A/S/D keys or up/ down/ left/ right arrow keys.	The character is able to move forward, backward, left and right	/
2	Player character can jump	Press the spacebar on the keypad.	The character can jump.	/
3	Player character can speed up the movement	Press W/A/S/D together with Left Shif key	Player character can speed up the movement	/
4	Player character can shoot	Click the left key of the mouse button.	The character can shoot.	/
5	Player character can collect items	Walk near the items	The items will disappear and keep as the ammo of the weapon.	/
6	The Germs can attack the player character.	The player character goes near the Germ.	The Germs attack the player character.	/
7	Germs in the gameplay can drop collectable health item or ammo box.	The player character shoots down the germ.	Germs in the gameplay drop ammo box or ammo box.	/
8	Points can be added.	The player character shoots the items to the bins correctly. Besides, points will be added after the player character shoots down the enemies successfully.	Points added.	/
9	Points can be added to Orange Bin	The player character shoots the plastic bottles ammo or aluminium cans to the Orange bin	10 points added for 1 bottles ammo shot to the Orange Bin	/
10	Points can be added to Brown Bin	The player character shoots	10 points added for 1 bottles ammo shot	/

		the glass bottles ammo to Brown Bin	to the Brown Bin.	
11	Points can be added to Blue Bin	The player character shoots the dirty boxes and cardboard ammo to Blue Bin	10 points added for 1 dirty boxes and cardboard ammo shot to the Blue Bin.	/
12	Collectable ammo box drops from the Orange bin, Blue Bin and Brown Bin.	The player character shoots the plastic bottles, glass bottles or boxes of ammo to the Orange bin, Brown Bin and Blue Bin to earn points respectively. Every 100 points drop a ammo box for all Bins.	Collectable ammo box drops from the Orange bin, Blue Bin and Brown Bin.	/
14	Timer can count down from 900 seconds	Once the player confirm to start game	Timer start counting down from 900 seconds	/
15	Game Over Canvas shown	Once player is dead or times up	Game Over Canvas shown	/
16	Victory Canvas	Once player complete all the mission	Victory Canvas shown	/

Test Case 8: Pause Menu				
No.	Functionality	Pre-Condition	Expected Outcome	Pass(/) or Fail (X)
1	Resume the game.	Players press the pause button (Esc key) when the game is paused.	The game will be resumed when the pause button is pressed when the game is pausing. The pause menu will be hidden away.	/
2	Restart the level.	Players press the restart button.	The level will be restarted from the	/

			start.	
3	Load the main menu	Players must first pause the game and then press the main menu button.	The current progress of the level will be lost and the game brings the player back to the main menu scene.	/
4	View Control/Help Panel	Player press the Control/Help button	Control/Help Panel shown	/

APPENDIX B

SURVEY FORM

Evaluation Form

Dear valued respondents,

I'm Goh Mei Kei from the Faculty of Computing, University Malaysia Pahang(UMP). I am currently conducting an evaluation form to get the valuable feedbacks from you on the Green Genie 3D edutainment game experience. It is an environment game that uses an indirect way to prompt the young generation to think about the importance of taking care of our environment. Recycling is one of the ways to protect our environment from waste pollution.

The download link for Green Genie: <https://meikei.itch.io/green-genie>

I really appreciate your time and cooperation to provide your response and feedback based on the following questions. Thanks in advance for your participation!

 meikeigoh@gmail.com (not shared) [Switch accounts](#)



*Required

What is your highest education level?

- Primary Level
- Secondary Level (Secondary School, High School)
- Tertiary Level (Colledge, University)
- Working

Does respondent think Green Genie's gameplay is fun and interesting?

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

The gameplay of Green Genie is easy to understand?

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Green Genie is suitable for elementary students to play?

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Green Genie has enhanced the recycling knowledge?

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Green Genie has raised the environmental awareness?

	1	2	3	4	5	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

Which Level of Green Genie do you think is the most difficult to pass? *

- Level 1: Trash Island
- Level 2: Trash Fall
- Level 3: Dead Tree Island
- Level 4: Trash City 1
- Level 5: Trash City 2

Kindly leave a short comment(s) on this game.

Your answer _____