

SIMULATION GAME FOR ROOM DESIGN

TAN SHAO KANG

Bachelor of Computer Science (Graphics &
Multimedia Technology) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : TAN SHAO KANG

Date of Birth

Title : SIMULATION GAME FOR ROOM DESIGN

Academic Session : SEMESTER II 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:

Suraya
(Supervisor's Signature)

New IC/Passport Number
Date: 11/7/2023

Name of Supervisor
Date: 24/7/2023

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: 24/7/2023

Stamp: **TS. DR. SURAYA ABU BAKAR**
SENIOR LECTURER
FACULTY OF COMPUTING
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG

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 bachelor degree of Computer Science (Graphics & Multimedia Technology) with honours.

(Supervisor's Signature)

Full Name : **TS. DR. SURAYA ABU BAKAR**
Position : **SENIOR LECTURER**
Date : 24/7/2023 **FACULTY OF COMPUTING**
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG



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.

(Student's Signature)

Full Name : TAN SHAO KANG

ID Number : CD19128

Date : 12 JUNE 2023

SIMULATION GAME FOR ROOM DESIGN

TAN SHAO KANG

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

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JUNE 2023

TABLE OF CONTENT

Table of Contents

TABLE OF CONTENT	5
LIST OF TABLES	7
LIST OF FIGURES	8
CHAPTER 1	11
INTRODUCTION	11
1.1 INTRODUCTION	11
1.2 PROBLEM STATEMENTS	12
1.3 OBJECTIVE	13
1.4 SCOPE	14
1.5 SIGNIFICANCE OF PROJECT	14
1.6 THESIS ORGANIZATION	14
CHAPTER 2	16
LITERATURE REVIEW	16
2.1 INTRODUCTION	16
2.2 REVIEW OF EXISTING APPLICATION	16
2.2.1 Game Application 1 - House Flipper	16
2.2.2 Game Application 2 - Cities: Skylines	19
2.2.3 Game Application 3 - The Sims 4	20
2.3 LITERATURE REVIEW (WORK ANALYSIS)	23
2.3.1 Analysis of comparison on previous system/method	23
CHAPTER 3	26
METHODOLOGY	26
3.1 INTRODUCTION	26
3.2 PROJECT MANAGEMENT FRAMEWORK	26
3.2.1 Planning	27
3.2.2 Design	27
3.2.3 Develop	28
3.2.4 Test	28

3.2.5 Deploy	28
3.2.6 Review	28
3.2.7 Launch	28
3.3 FLOW CHART	29
3.4 CONTEXT DIAGRAM	35
3.5 USE CASE DIAGRAM	36
3.6 STORYBOARD	37
3.7 SOFTWARE AND HARDWARE REQUIREMENT	49
3.8 GANTT CHART	50
3.9 TESTING	50
CHAPTER 4	53
IMPLEMENTATION, RESULTS AND DISCUSSION	53
4.1 INTRODUCTION	53
4.2 DEVELOPMENT TOOLS	53
4.3 IMPLEMENTATION	53
4.3.1 Main Menu Scene	54
4.3.2 Level Scene	55
4.4 CODING	59
4.5 RESULT AND DISCUSSION	59
CHAPTER 5	63
CONCLUSION	63
5.1 INTRODUCTION	63
5.2 LIMITATION AND CONSTRAINT	63
5.3 FUTURE WORK	64
REFERENCE	66
APPENDIX A	67
APPENDIX B	68
APPENDIX C	73
APPENDIX D	82

LIST OF TABLES

Table 1.1 Summary of problems	13
Table 2.1 the comparison of 3 existing system based on their feature/system	25
Table 3.1 Main Menu Scene	37
Table 3.2 Game Module Screen	38
Table 3.3 Demo Screen	39
Table 3.4 1-Bed Room Screen	41
Table 3.5 2-Bed Room Screen	44
Table 3.6 3-Bed Room Screen	46
Table 3.7 Double Storey Bed Room Screen	49
Table 3.8 List of software requirement	50
Table 3.9 List of hardware requirement	50
Table 3.10 User Acceptance Test (UAT) form	52
Table 4.1 Development tools for game application	53
Table 4.2 Summary of user's comments	62

LIST OF FIGURES

Figure 2.1 Game Logo of House Flipper	17
Figure 2.2 Example of game space	17
Figure 2.3 changing tools function in House Flipper	18
Figure 2.4 the task and the object interact UI	18
Figure 2.5 Game Logo of Cities: Skylines	19
Figure 2.6 Game space and the build space of the game	19
Figure 2.7 Example of UI of Cities: Skylines	20
Figure 2.8 Game Logo of The Sims 4	21
Figure 2.9 Game UI of The Sims 4	21
Figure 2.10 Inventory of The Sims 4	22
Figure 3.1 the process of Agile Model	27
Figure 3.2 Flowchart of Room Design game application	29
Figure 3.3 Flowchart of 1-Bed Room module	30
Figure 3.4 Flowchart of 2-Bed Room module	31
Figure 3.5 Flowchart of 3-Bed Room module	32
Figure 3.6 Flowchart of Double Storey Room module	33
Figure 3.7 Flowchart of Demo module	34
Figure 3.8 the Context diagram	35
Figure 3.9 Use case diagram	36
Figure 4.1 Main Menu Scene	54
Figure 4.2 Level Scene	54
Figure 4.3 Instruction of the gameplay	55
Figure 4.4 Game Scene	55
Figure 4.5 Inventory Scene	56
Figure 4.6 Object detected	56
Figure 4.7 Object moving	57
Figure 4.8 object deselected	57
Figure 4.9 Game menu	58
Figure 4.10 the UI of view mode	58
Figure 4.11 the picture saved location	59

Figure 4.12 survey on game UI	60
Figure 4.13 survey of game experience to the user	61
Figure 1 Gantt chart of the project	67
Figure 2 Usability test question using Google Form (Section 1)	68
Figure 3 Usability test question using Google Form (Section 2)	69
Figure 4 Usability test question using Google Form (Section 3)	70
Figure 5 Usability test question using Google Form (Section 4)	71
Figure 6 Usability test question using Google Form (Section 4)	72
Figure 7 UAT Form result for 1 st person	73
Figure 8 UAT Form result for 1 st person	74
Figure 9 UAT Form result for 1 st person	75
Figure 10 UAT Form result for 2 nd person	76
Figure 11 UAT Form result for 2 nd person	77
Figure 12 UAT Form result for 2 nd person	78
Figure 13 UAT Form result for 3 rd person	79
Figure 14 UAT Form result for 3 rd person	80
Figure 15 UAT Form result for 3 rd person	81
Figure 16 Script for play the game	82
Figure 17 Script for open the option	82
Figure 18 Script for Option	82
Figure 19 Script for quit the game	83
Figure 20 Script for load the module	83
Figure 21 Script for display the game instruction when in-game	84
Figure 22 Script for middle the mouse position to the screen and player control	86
Figure 23 Script for open the Inventory and Menu UI	87
Figure 24 Script for spawn the object	88
Figure 25 Script for select the object	90
Figure 26 Script for drag and move the object	90
Figure 27 Script for rotate the object	90
Figure 28 Script for delete the object	90
Figure 29 Script for move, rotate and zoom the camera in view mode	93
Figure 30 Script for take the screenshot	93

LIST OF ABBREVIATIONS

2D	2-Dimensional
3D	3-Dimensional
UI	User Interface
UAT	User Acceptance Test
OBJ	Object
PC	Personal Computer

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In 1950s, the first game prototype has been created and in 1971 first consumer arcade video game “Computer Space” has released. That's when games started to take off in our lives. Now games have become a part of our lives, and games have brought us a lot of influence. Some people like to play games, some people like to develop games and some people use games to transmit the information. Modern game need a device/console to play it, for example PC, Smartphone, PlayStation console etc. and for game develop we need use the engine to develop the game because the game is a virtual thing.

There are many type of game has been released, the game can be clarify by game genre. The genre of games are Action, Fighting, Adventure, Platform, Puzzle, Racing, Role-playing, Simulation, Shooter, Sports, Strategy and Misc. Game genre are classified by game content, gameplay, and physical equipment. Game also can be clarify by these two type, Entertainment game and serious game. Entertainment game is for playing for fun, the game content is not so important. Serious game is for transmit some message to the player or with some aim. So the game content and game element are very important when develop a serious game.

In this project, the architecture simulation game is develop for play for fun purposes. With this simulation game the user can find inspiration of the architectural setting and also can gain the experience. This game can let the user learn the design and also can help inspire creativity of the user. This game have 2 module; City and Room. User can choose one of them to design. After the

user satisfy with their design, they can choose to save the design to the device, print out the design or share the design to others.

1.2 PROBLEM STATEMENTS

The aim for this project is to develop a simulation game to let the user learn the design and inspire the creativity of user. When we need to design something like our home we need to know how to design first. If we didn't have the knowledge of design we can only ask others to help design, which will cost a huge amount of money. Not only that, if we got any idea of design we need something to present our design, sure we can draw it on a paper but the result maybe it's not that good.

After that, nowadays people are busy with work and study so if someone that are interest with design is very difficult to inspire their design creativity. If using the professional application like AutoCAD or SketchUp is very hard to do it because professional application needs some basic knowledge to use it. With game application user can inspire their design creativity while playing the game.

Next, when we done our design we want to show our design to other person or architect. If we draw our design on a paper it is difficult to let other person to understand our design that we want unless that person knows design very well. So with the game we have a virtual design item to give other person know how our design looks like and other person also can easily to understand our design that we want.

Lastly, to do the design we mostly use the application to do it. Most of the design application require knowledge about the application before we can use the application. We need to spend a lot of time to get familiar with how to use the application so maybe some of the person will feeling tired to learn it and give up. So with game application it is easy to use it. Game application is simple to do the design and the user also can have fun when doing the design. Designing in a game application takes less time and cost than using a design app.

Problem No	Problem	Description	Effect
1	Lack of knowledge about design	Most of the user don't understand the design.	will degrade a person's sense and vision
2	Difficult to inspire design creativity of user	Some of the design application hard to help the user inspire their design creativity because the requirement of the design application are too high	If user unable to inspire their design creativity their idea will outdate and less
3	Difficulty showing the design in the mind of the user	To present the design we need to do it first in an application and show to others people. If a person didn't know how to use design application it is hard to show the design to others people.	Can't transmit the design idea to the others people when doing the design
4	Design application are difficult to use	Design application need some basic knowledge to use it and some design application have function limit to the user to do the design unless the user buy their application	This will cause user feel difficult to do the design and will give up

Table 1.1 Summary of problems

1.3 OBJECTIVE

The objectives of the project are:

- i. To study the existing architecture simulation games
- ii. To design and develop Room Design game for students
- iii. To evaluate the performance and the functionalities of the purpose games

1.4 SCOPE

There are 3 types of scope in this project which are user scope, system scope and development scope. The user scope is the target user of the game. The target user for the game is for student age between 7 to 18 years. System scope is this game are a game that combine with 2D and 3D. The genre of this game is simulation game and this game have 4 module which is 4 different room that will be given. The game allow user to save the picture of their design. Development scope is this game using unity, Photoshop and some multimedia element like text, audio, graphics, animation to develop it.

1.5 SIGNIFICANCE OF PROJECT

The significance of this project is to develop an interest in design to the people and also can training design ability and idea. Design is very important to us because most things in our lives involve design so if people are feel interest in design it will contribute to the development and planning of the country. Other than that with design we also can create something new from existing thing.

1.6 THESIS ORGANIZATION

This thesis consists of five chapters. Chapter 1 explained about the introduction to the project, which are the problem statements, the objectives, the scope, significance of the project and report organization.

Chapter 2 is the literature review of three existing game application. In this chapter the analysis of existing game application will be done and make the comparison of these existing game application.

Chapter 3 explained about the methodology will use to develop the project. All of the development process like framework, requirement, design, testing and solution will be explained in this chapter.

Chapter 4 explained the development, result and discussion of this project. The results will be discussed in this chapter.

Lastly is chapter 5, in this chapter all the result of the project will be summarized and the limitation and constraint of this project will be discussed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is about the analysis and review of existing game application in simulation. There are 3 existing simulation game will be analysis and do the comparison to find out the features, strength, weakness, advantages and disadvantages. The analysis and comparison will focus on Graphical User Interface (GUI), Operating System (OS), game genre, technique, language provided, target audiences, size of application, functions provided and tools or technology to ensure the game develop for this project is better than existing game.

2.2 REVIEW OF EXISTING APPLICATION

Three existing simulation game will be reviewed. These three applications are House Flipper, Cities: Skylines and The Sims 4

2.2.1 Game Application 1 - House Flipper

House Flipper is a simulation game that buy, repair and remodel devastated houses. In this game user will learn how to repair the house with the tools and remodel the house with their design. This game available on Mobile, PC, PS, and Xbox. Games can be purchased on the above platforms.

There are some DLC in this game, the DLC can let the user change the environment. The DLC environment have Garden, Pets house, Farm, Cyberpunk city and Luxury house. User can choose a house to clean and renovate it. After the user done clean and renovate user can sell the house to earn the profit.



Figure 2.1 Game Logo of House Flipper

This game application is a 3D game and its purpose is play for tidy up the interior. In this game the player can place the furniture in the map and also can clean the house by changing the tools.



Figure 2.2 Example of game space



Figure 2.3 changing tools function in House Flipper

This game have Task for the user to do it to earn the income. The income in this game can buy the furniture and place in the map. All of the object in this game can be interact to put and rotate the object.



Figure 2.4 the task and the object interact UI

2.2.2 Game Application 2 - Cities: Skylines

Cities: skylines is a city-building game. In this game user can planning design their urban development. This game available on PC, Xbox, PS and Nintendo Switch. This game also need to purchase it to play.

In this game user will be given a plot of land (1.2 mi × 1.2 mi) area. User need to use the budget given to design and develop the city.



Figure 2.5 Game Logo of Cities: Skylines

This game is a 2.5d game. The purpose of this game is to build a city. In this game a wide map will be given and the player can click on the map to check the build space for the building.



Figure 2.6 Game space and the build space of the game

There are many type of building in this game that player can build it but player need to gain the money to buy it on the building menu.



Figure 2.7 Example of UI of Cities: Skylines

2.2.3 Game Application 3 - The Sims 4

The Sims 4 is a social simulation game. In this game user can design the character before enter the game, after user enter the game user can design their house. This game available on PC, PS and Xbox. This game need to purchase first before play but this game have demo version for the user to test play the game.

In this game there is no primary objective or goal to achieve. The main objective of this game is let the user simulate their life in this game. This game have a mode that call build mode. In this mode user can design their house more efficiency. User also can share their house design to other people.



Figure 2.8 Game Logo of The Sims 4



Figure 2.9 Game UI of The Sims 4

Player can select the house furniture in the inventory and place in the map. This player need to gain money and EXP to unlock more furniture in the inventory.



Figure 2.10 Inventory of The Sims 4

2.3 LITERATURE REVIEW (WORK ANALYSIS)

2.3.1 Analysis of comparison on previous system/method

Application	House Flipper	Cities: Skylines	The Sims 4
Graphical User Interface (GUI)	The GUI is good, the light effect, the texture are close to reality	The GUI are large and got some chaos because too many element in the game. The light effect and the texture are good.	The GUI is simple but the function is very complete. The light effect and the texture are good.
Operating System (OS)	PC, Android, IOS, PS, Xbox, Linus, MacOS	PC, PS, Linux, Xbox, Nintendo, MacOS	PS, Xbox, PC
Game Genre	Simulation	Simulation	Simulation
Game Mode	Single-player	Single-player	Single-player
Language Provided	20 language (English, Chinese, Japanese, Korean, French...)	9 language (English, Chinese, German, Korean, French...)	18 language (English, Chinese, Japanese, Korean, French...)
Target User	Kids 3 age above, student and teenager interest with building design	Kids 3 age above, student and teenager interest with building design	Kids 12 age above, student and teenager
Size of Application	6 GB	4 GB	17 GB
Function	<ul style="list-style-type: none"> • Have start game and continue game • Can save the game progress 	<ul style="list-style-type: none"> • Have start game and continue game • Can save the game progress 	<ul style="list-style-type: none"> • Have start game button • Can design the character • Have gallery to save the gameplay picture

	<ul style="list-style-type: none"> • Can change the game settings • Have button to exit the game • Have minimap show where the player at • Has shown the task and the progress 	<ul style="list-style-type: none"> • Can change the game settings • Have button to exit the game • Can choose the map to play • Have tips to help the user how to play the game 	<ul style="list-style-type: none"> • Can choose the map to play • Have a UI to help the user select the tools or item easily
Advantages	<ul style="list-style-type: none"> • User will know the function of tools and item • User can learn how to design a house • This game will show the detail how house equipment be installed 	<ul style="list-style-type: none"> • Users can learn how to design cities in detail • User will learn the importance of urban development • This game has involve ecosystem so the user also can learn the ecosystem when playing this game 	<ul style="list-style-type: none"> • User can learn the house design • There are many type of item user can use in design • The game perspective is better to see whole the house design • Can refer to other people's designs

<p>Disadvantages</p>	<ul style="list-style-type: none"> • Game does not have main objective so it didn't notice user what they should do in this game. User need to find out themselves. 	<ul style="list-style-type: none"> • The map is very big and very difficult to design the city • This game has many elements that make the game a bit complicated and difficult to play • The building build not so detail 	<ul style="list-style-type: none"> • The Game Space of this game using the box so the space has limit • Game elements are more biased towards life simulation, so the design element will not too much
-----------------------------	--	---	--

Table 2.1 the comparison of 3 existing system based on their feature/system

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter is explained about the methodology, requirement, planning and the design of the project. In this chapter the detail of the methodology that use to develop the project like SDLC and Research framework will be explain. After that, the project requirement and user requirement will be determined to develop the project. After have the requirement, the Gantt chart for planning and schedule, and the data design, interface design will be done. The design of testing plan and validation plan will also be done in this chapter.

3.2 PROJECT MANAGEMENT FRAMEWORK

The management framework that use in this project is Agile models. The reasons that I use this model to manage my project because with this model I can easily to view the whole process of the development and also can spot the problem quickly. Other than that, with this model I can do the improvement of the project during development because with this model I can easily to make the changes.

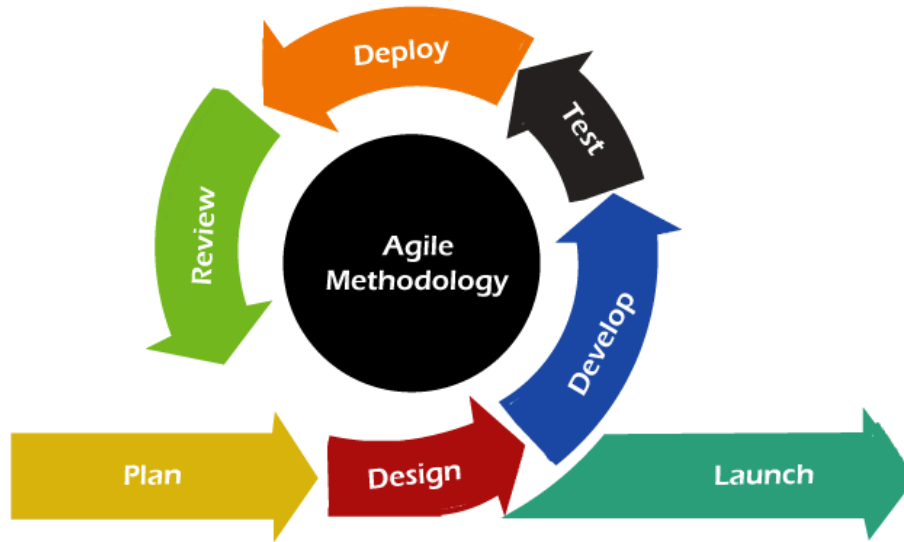


Figure 3.1 the process of Agile Model

3.2.1 Planning

Planning phase is an early phases before develop a project. Planning phase will brainstorming the project that suitable to meet our objective. Objective is important in a project because it is a goal of the project. In this phase the project scope will be determined. The target user is for the student age 7 to 18 years. This project will be develop in 2D and 3D PC platform game and the game is about design learning. The software that use to build this project are Unity, Microsoft Visual Studio 2019 and Adobe Photoshop. Multimedia element such as text, image and audio will also add in the game.

3.2.2 Design

Design phase is where all the game content will be created. In this phase the game content will be design by using Adobe Photoshop and import in unity for develop process. The design of the game content are based on the objective and the user scope.

3.2.3 Develop

Develop phase will start once the design phase already done. In this phase all the multimedia content will be import to Unity for develop the game. The software that use in this phase are Unity and Microsoft Visual Studio 2019. Unity is used to create a game space and gameplay of the game application. Visual Studio 2019 is used to scripting the game function.

3.2.4 Test

Testing phase is to test the game application once it's done. In this phase the game functionality and all game features will be test by some user. User Acceptance Test (UAT) is used to test the game application.

3.2.5 Deploy

Deploy phase will be done based on the User Acceptance Test (UAT) from previous phase. Any changes and improvement on the game application will be done and the problem, error or bug will be fixed.

3.2.6 Review

In this phase Usability Test form for the game application will be created and this form will share to others people to get the feedback.

3.2.7 Launch

The game application will be built for playing. The game application will be uploaded in App Store, Website or the drive for user and client to launch it.

3.3 FLOW CHART

Flow chart is a workflow of the gameplay for game application. Figure 3.2 shows the overall game flow of the game application. Figure 3.3, 3.4, 3.5, 3.6 and 3.7 show the game flow of each module.

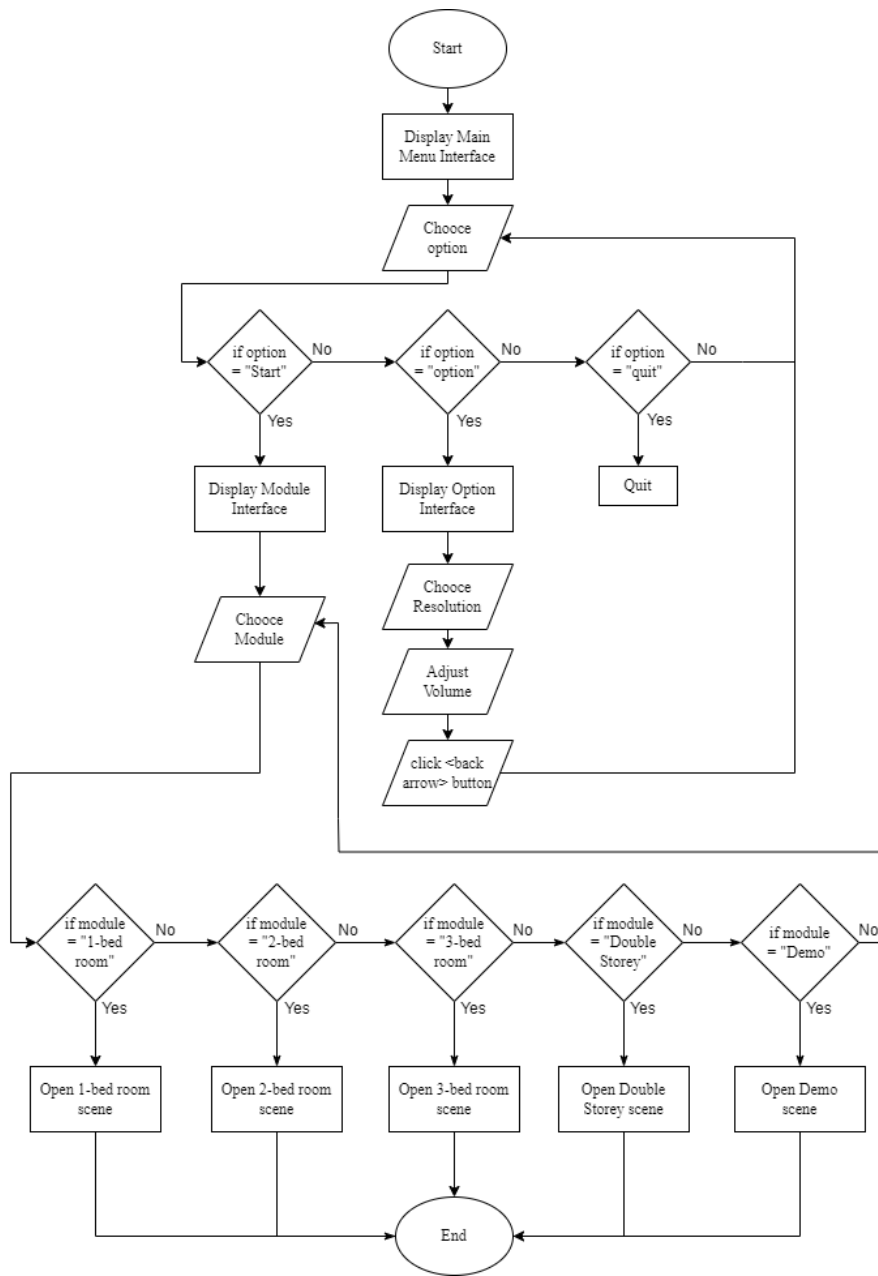


Figure 3.2 Flowchart of Room Design game application

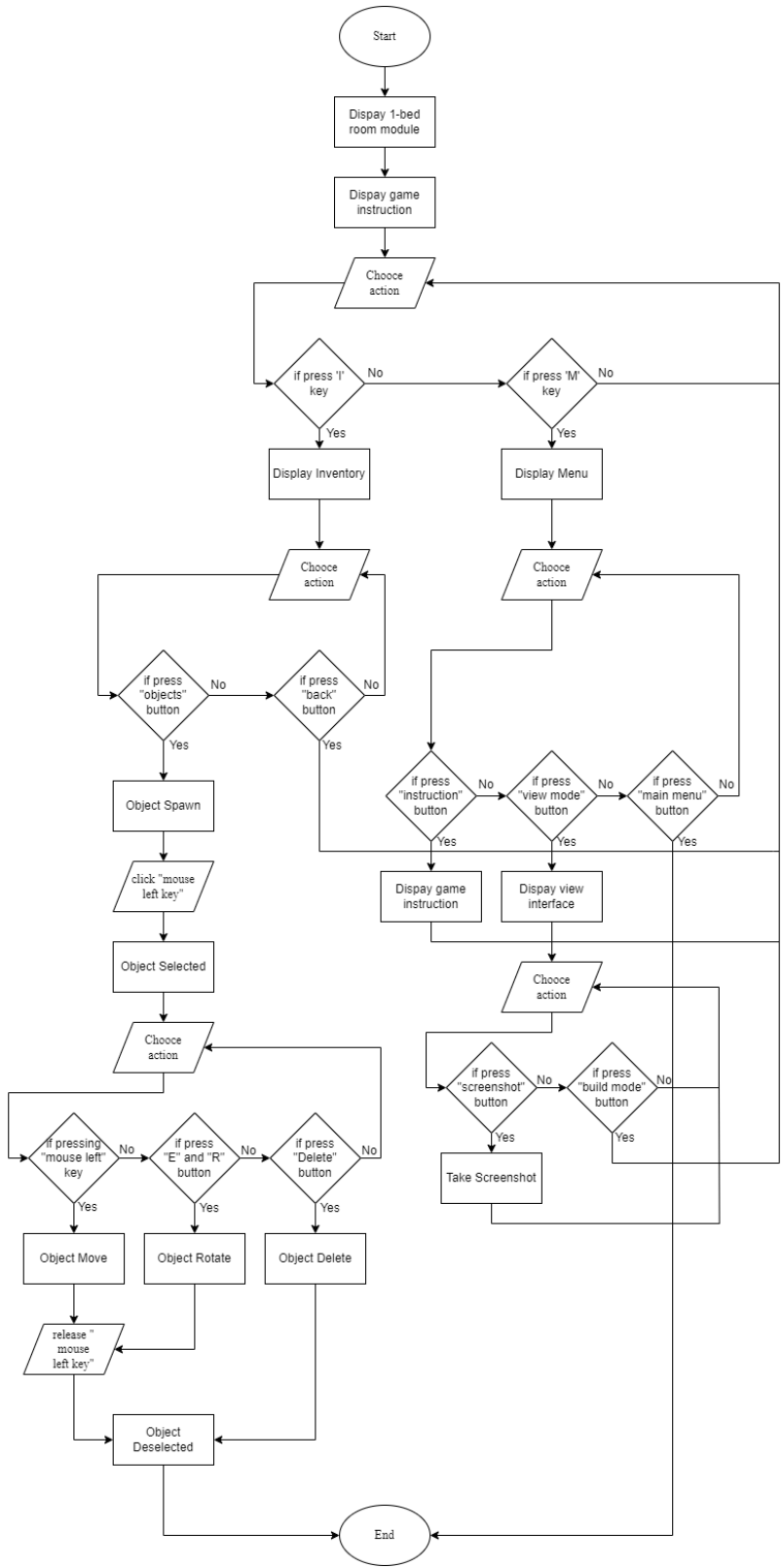


Figure 3.3 Flowchart of 1-Bed Room module

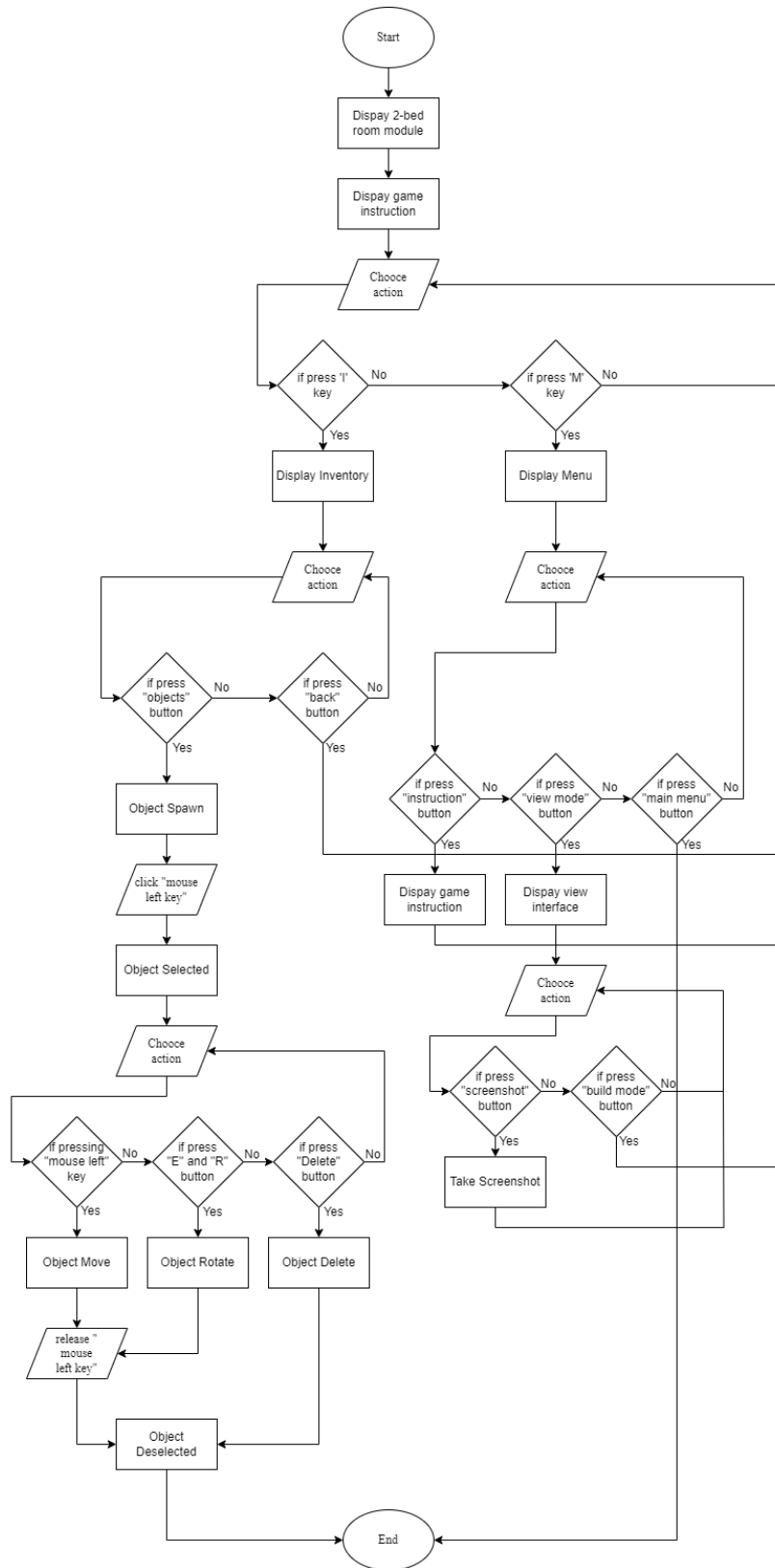


Figure 3.4 Flowchart of 2-Bed Room module

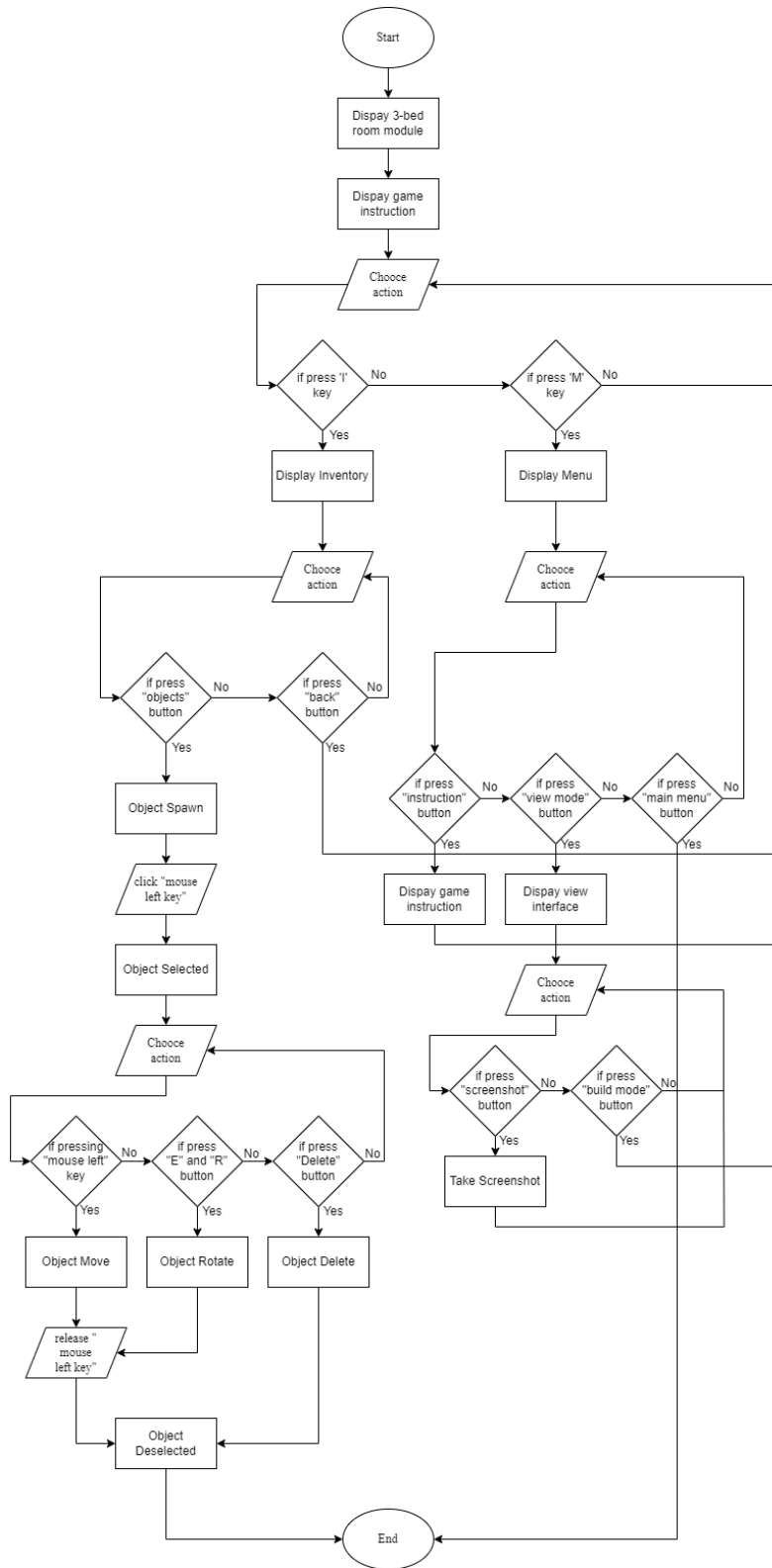


Figure 3.5 Flowchart of 3-Bed Room module

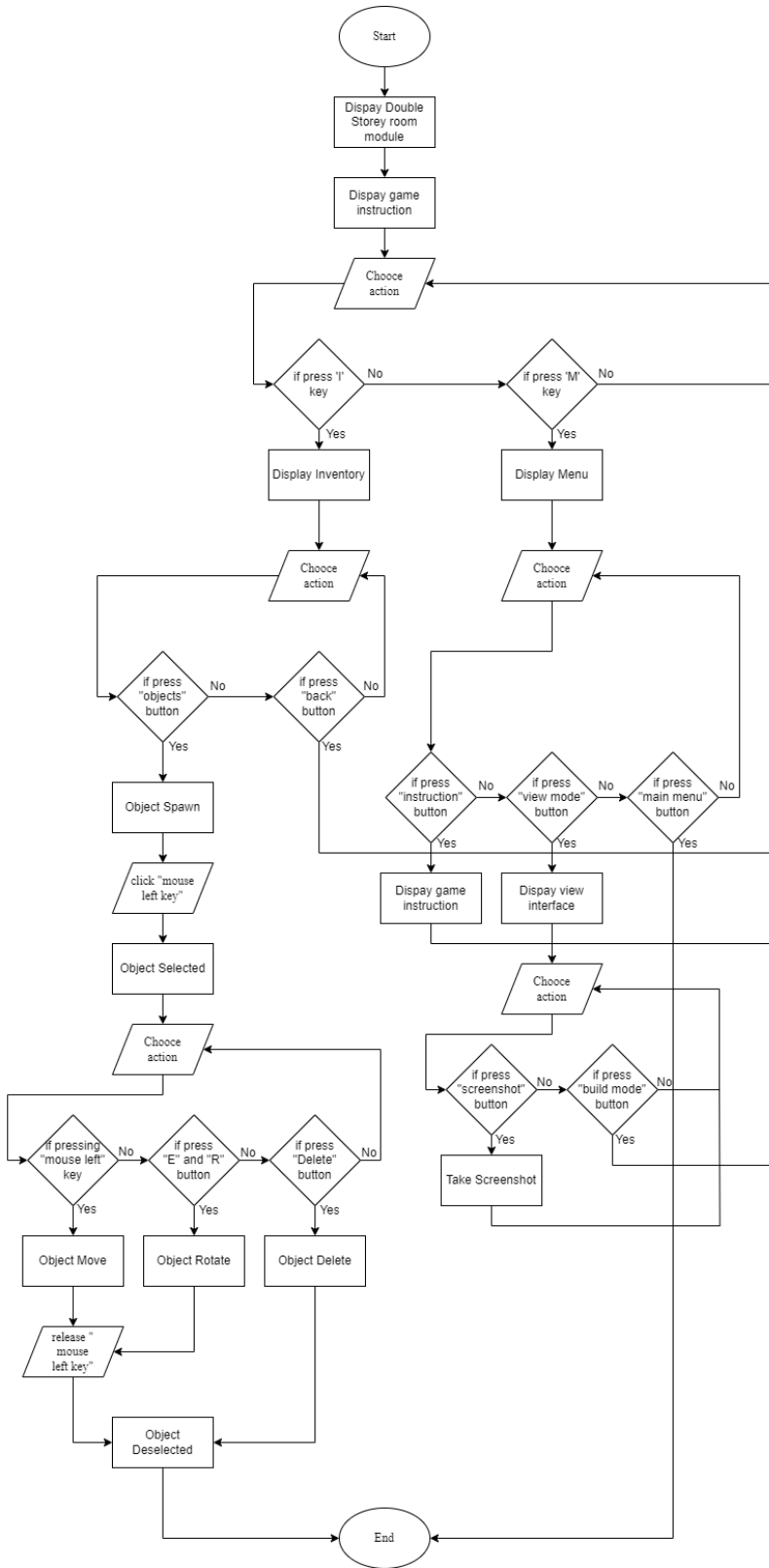


Figure 3.6 Flowchart of Double Storey Room module

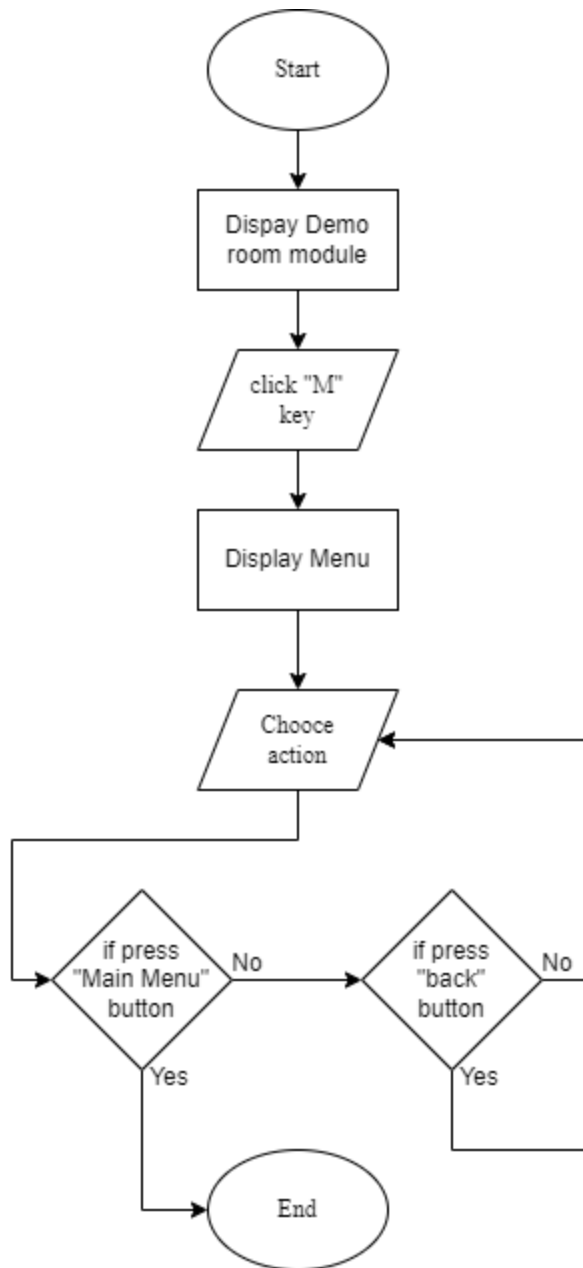


Figure 3.7 Flowchart of Demo module

3.4 CONTEXT DIAGRAM

Context diagram is used to show the relationship of the game application. This game application have 3 entities which is Player, System and Publisher. The Player will run the game application to play the game and provide the feedback. The Publisher will create the game and make the update to the game application based on the user's feedback. System will save all the player data and display the game data to the user.

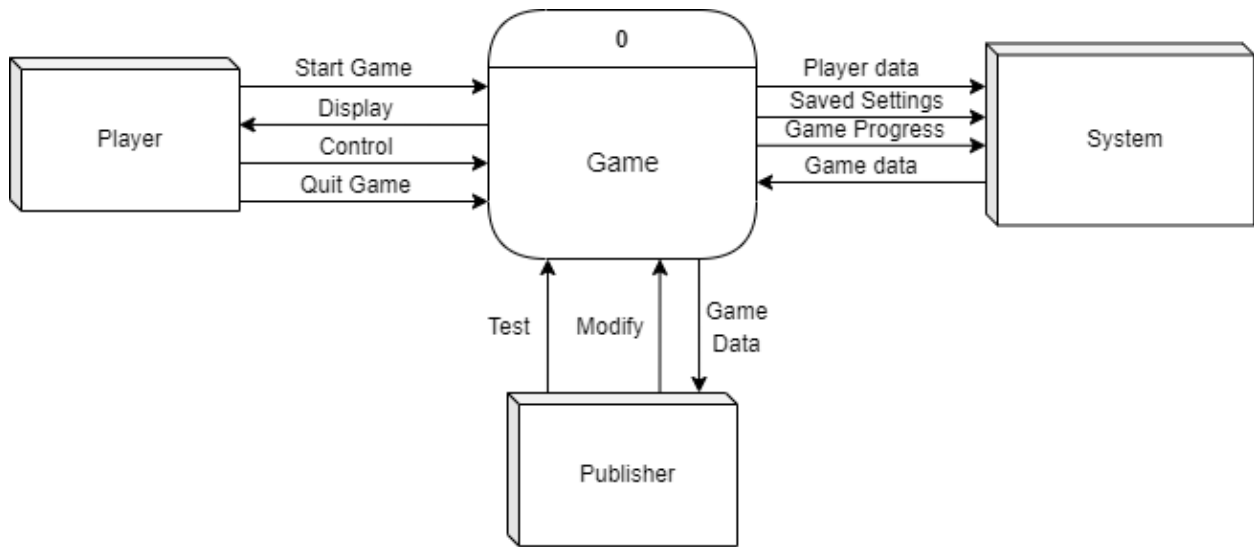


Figure 3.8 the Context diagram

3.5 USE CASE DIAGRAM

Use case diagram show the interaction between the user and the game application. Figure 3.9 show the use case diagram of Room Design game application. User are given 3 option in first screen. When user click on “Start” button the user will go to second scene that consist 4 module; 1-Bed Room, 2-Bed Room, 3-Bed Room and Double Storey Room to choose. In each module user can move around and spawn the object to design the map. If the user click on “Option” button the user can adjust the game volume and game resolution. If user click on “quit” button the game will close.

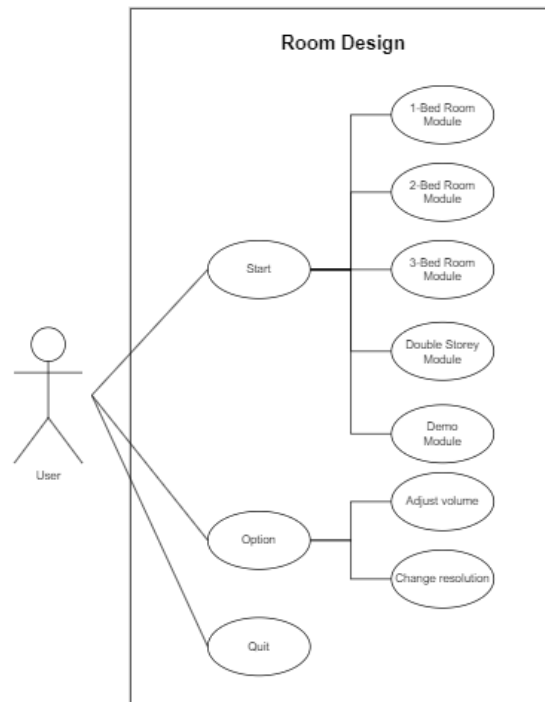


Figure 3.9 Use case diagram

3.6 STORYBOARD


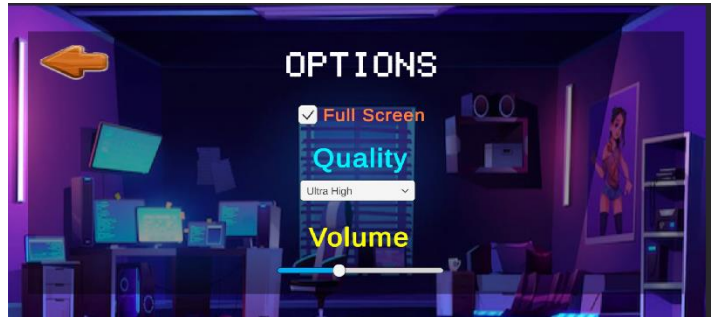


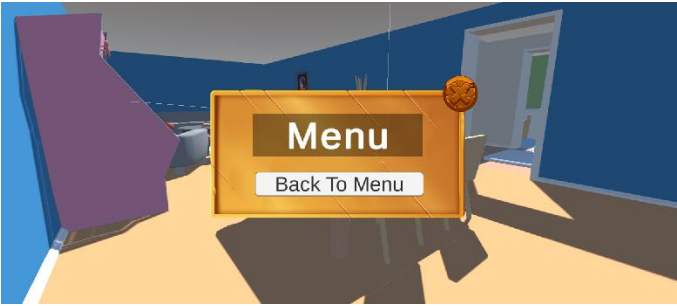
Screen Title : Game Main Menu	
Screen Layout	Description
	<p>This scene shows the main menu of Room Design. There are 3 button in the screen; “Play”, “Option” and “Exit”. Below are the function of these button:</p> <ul style="list-style-type: none"> • Play – Go to module screen • Option – Go to option screen • Exit – Close the game application
	<p>In option scene have a “←” button back to main menu. Full screen checkbox to enable or disable full screen when playing. Quality to set the game graphic and volume slider to adjust the game volume.</p>

Table 3.1 Main Menu Scene

Screen Title : Game Module	
Screen Layout	Description
	<p>This scene shows the module of Room Design. There are 4 module button in the screen and 1 “←” button and 1 “Demo” button. Below are the function of these button:</p> <ul style="list-style-type: none"> • ← – Go to main menu screen


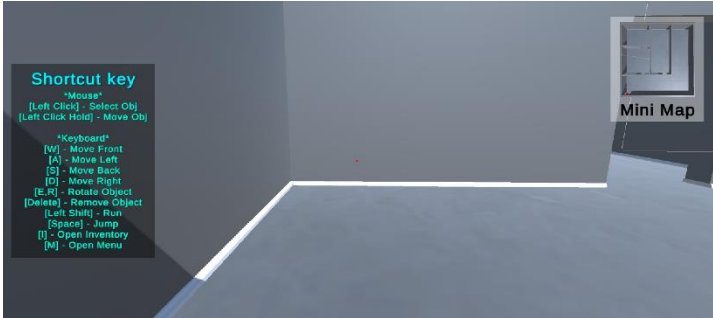
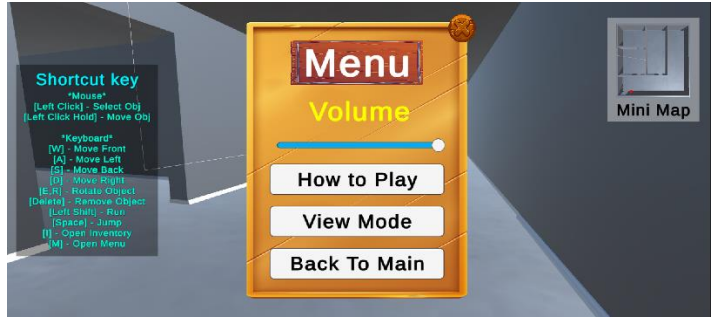
	<ul style="list-style-type: none"> • Demo – Go to demo screen • 1-Bed Room – go to 1-Bed Room scene • 2-Bed Room – go to 2-Bed Room scene • 3-Bed Room – go to 3-Bed Room scene • Double Storey Bed Room – go to Double Storey Room scene
--	--

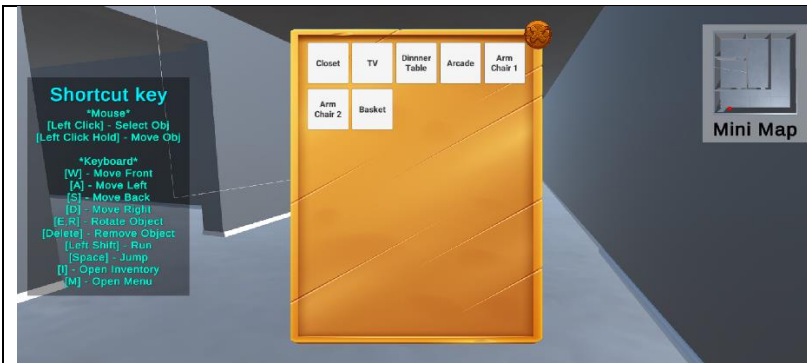
Table 3.2 Game Module Screen

Screen Title : Demo scene	
Screen Layout	Description
 	<p>This scene shows the demo module of Room Design. User can control the character move around in this scene. There are some key used in this scene. Below are the function of these key:</p> <ul style="list-style-type: none"> • WASD – Character move • Left Shift – increase character move speed • Space – Character jump • M – open menu <p>In Menu only have 2 button which is “Back to main menu” and “x” button. “Back to main menu” will go</p>

	back to main menu and “x” will close the Menu
--	---

Table 3.3 Demo Screen

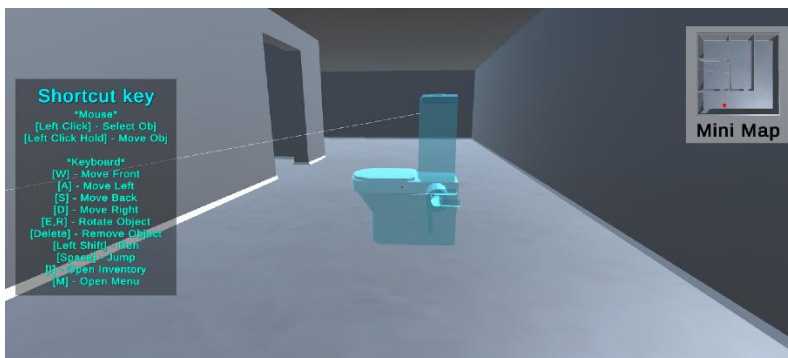
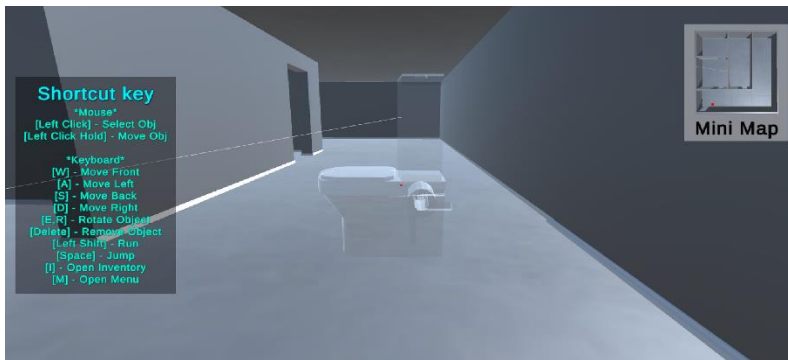
Screen Title : 1-Bed Room scene	
Screen Layout	Description
	<p>This scene shows the 1-Bed Room module of Room Design. When user entry this module a instruction will pop up and user can click “Next” button to skip it.</p>
	<p>User can control the character move around in this scene. There are some key used in this scene. Below are the function of these key:</p>
	<ul style="list-style-type: none"> • WASD – Character move • Left Shift – increase character move speed • Space – Character jump • M – open menu • I – open inventory • ER – Rotate object • Mouse left key click – select object



- Mouse left key hold – Move object

In Menu only have 4 button and 1 slider. Below are the functions of these items:

- Volume slider – to adjust the game volume
- “How to play” – to show the game instruction
- “view mode” – change the game mode
- “Back to main” – go to main menu

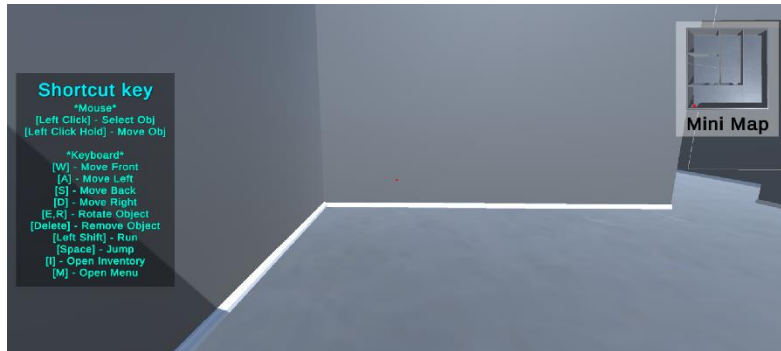
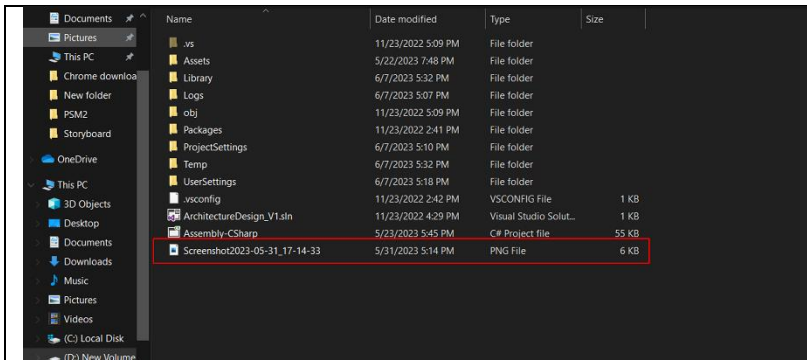


In Inventory there are some object button and 1 “x” button. Below are the function of the button:

- Object button – spawn the object
- x - close the Inventory

Each object button will spawn different object. The object will spawn in front of the user and user can select





the object to move, rotate and delete the object.

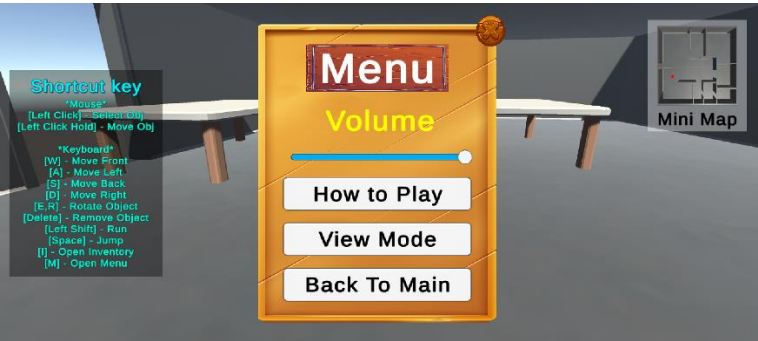
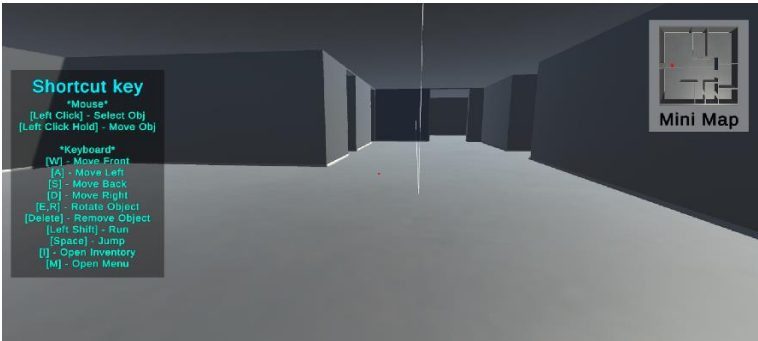
Once the user entry the view mode, there have some key use in this mode. Below are the key function:

- WASD – Move the camera
- QE – Rotate the camera
- RF – Zoom in/zoom out camera

There are 2 button in view mode which is “Screenshot” and “build mode”. Screenshot can take the picture of the screen and save in the game file. Build mode for go back to default mode.

Table 3.4 1-Bed Room Screen

Screen Title : 2-Bed Room scene	
Screen Layout	Description
	This scene shows the 2-Bed Room module of Room Design. When user entry this



module a instruction will pop up and user can click “Next” button to skip it.

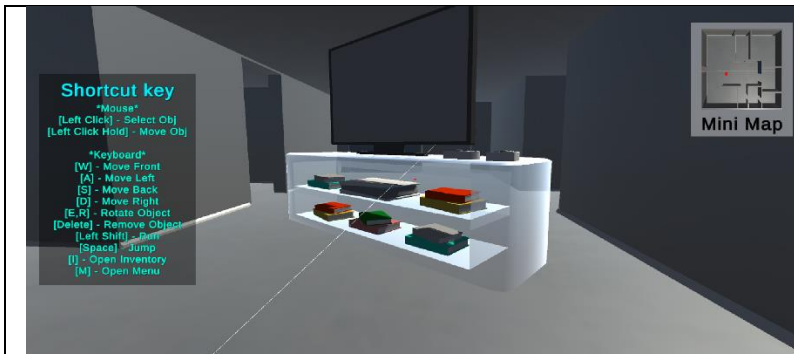
User can control the character move around in this scene.

There are some key used in this scene. Below are the function of these key:

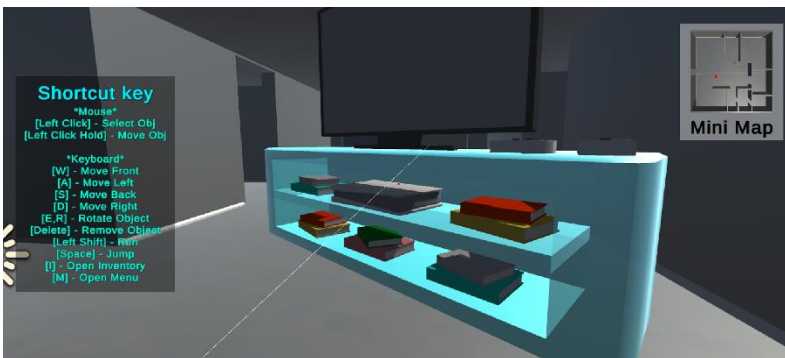
- WASD – Character move
- Left Shift – increase character move speed
- Space – Character jump
- M – open menu
- I – open inventory
- ER – Rotate object
- Mouse left key click – select object
- Mouse left key hold – Move object

In Menu only have 4 button and 1 slider. Below are the functions of these items:

- Volume slider – to adjust the game volume

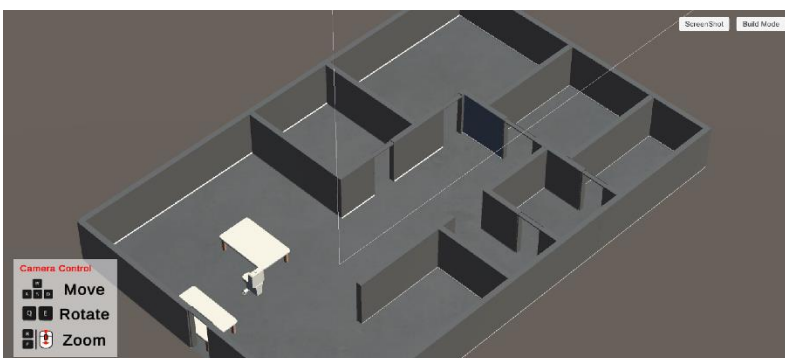


- “How to play” – to show the game instruction
- “view mode” – change the game mode
- “Back to main” – go to main menu

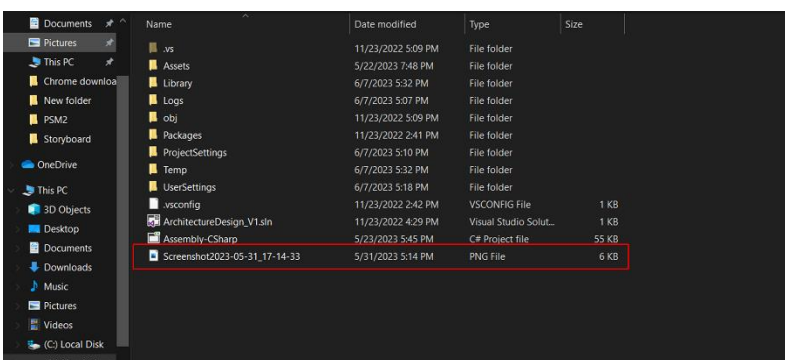


In Inventory there are some object button and 1 “x” button. Below are the function of the button:

- Object button – spawn the object
- x - close the Inventory



Each object button will spawn different object. The object will spawn in front of the user and user can select the object to move, rotate and delete the object.



Once the user entry the view mode, there have some key use in this mode. Below are the key function:

- WASD – Move the camera
- QE – Rotate the camera

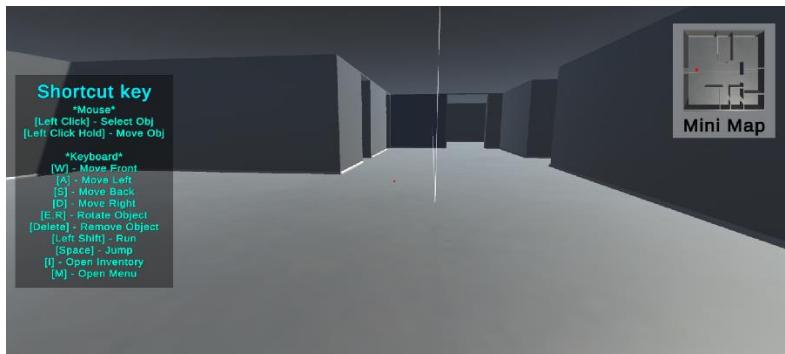

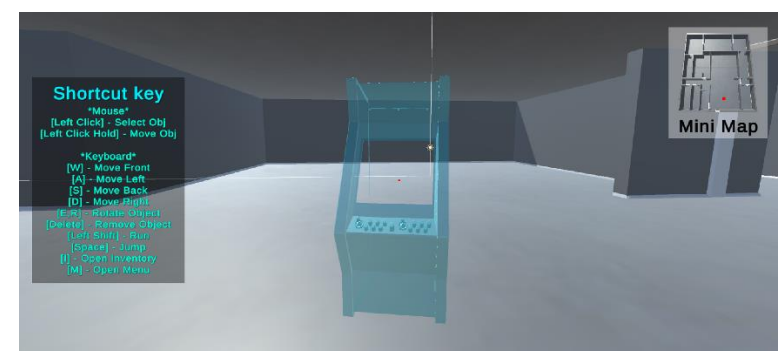
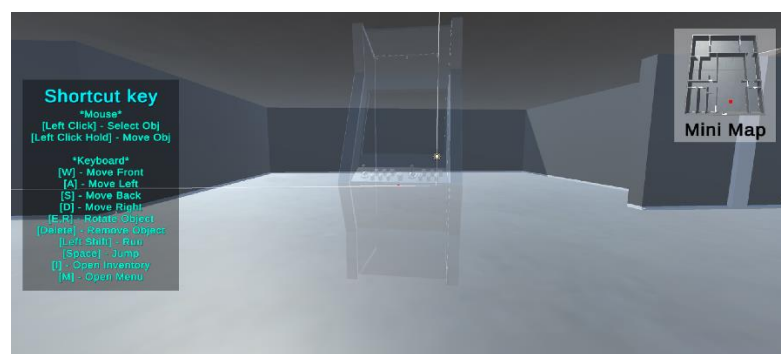
	<ul style="list-style-type: none"> • RF – Zoom in/zoom out camera <p>There are 2 button in view mode which is “Screenshot” and “build mode”. Screenshot can take the picture of the screen and save in the game file. Build mode for go back to default mode.</p>
---	--

Table 3.5 2-Bed Room Screen

Screen Title : 3-Bed Room scene	
Screen Layout	Description
	<p>This scene shows the 3-Bed Room module of Room Design. When user entry this module a instruction will pop up and user can click “Next” button to skip it.</p> <p>User can control the character move around in this scene. There are some key used in this scene. Below are the function of these key:</p> <ul style="list-style-type: none"> • WASD – Character move

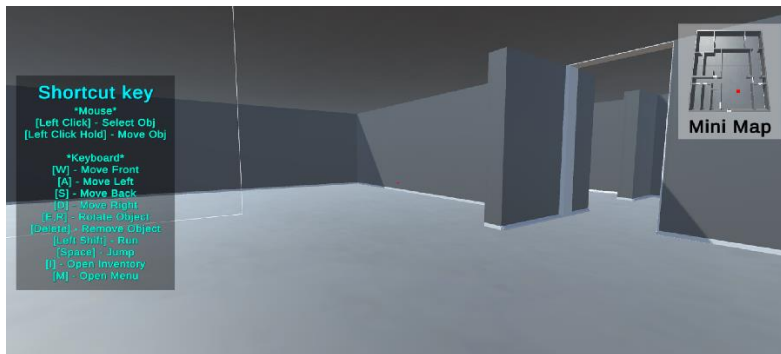
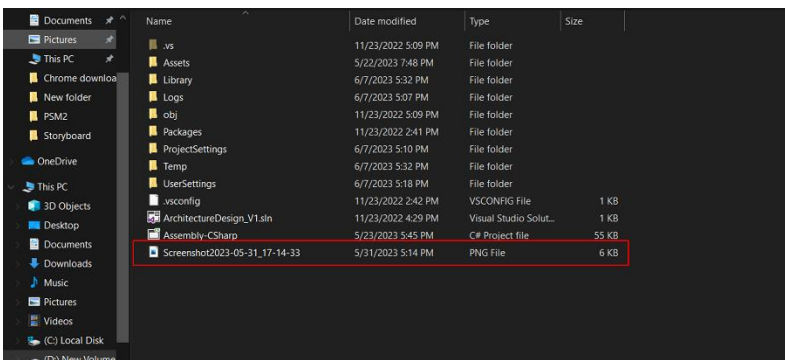


- Left Shift – increase character move speed
- Space – Character jump
- M – open menu
- I – open inventory
- ER – Rotate object
- Mouse left key click – select object
- Mouse left key hold – Move object

In Menu only have 4 button and 1 slider. Below are the functions of these items:

- Volume slider – to adjust the game volume
- “How to play” – to show the game instruction
- “view mode” – change the game mode
- “Back to main” – go to main menu

In Inventory there are some object button and 1 “x” button. Below are the function of the button:



- Object button – spawn the object
- × - close the Inventory

Each object button will spawn different object. The object will spawn in front of the user and user can select the object to move, rotate and delete the object.


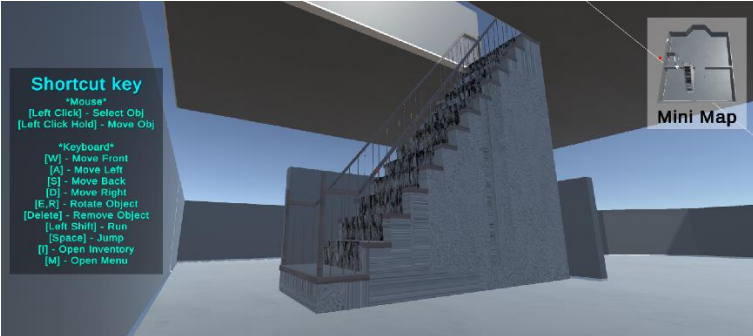

Once the user entry the view mode, there have some key use in this mode. Below are the key function:

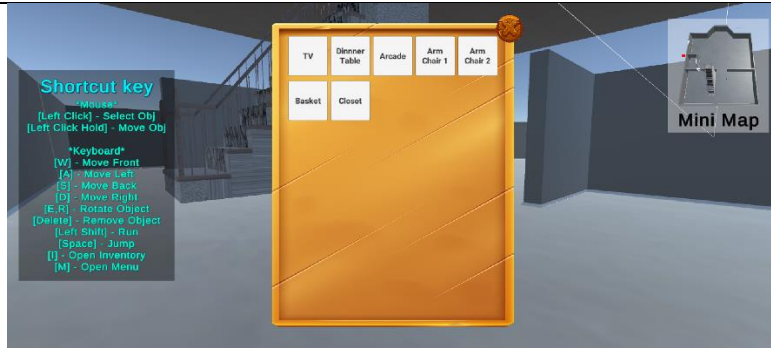
- WASD – Move the camera
- QE – Rotate the camera
- RF – Zoom in/zoom out camera

There are 2 button in view mode which is “Screenshot” and “build mode”. Screenshot can take the picture of the screen and save in the game file. Build mode for go back to default mode.

Table 3.6 3-Bed Room Screen

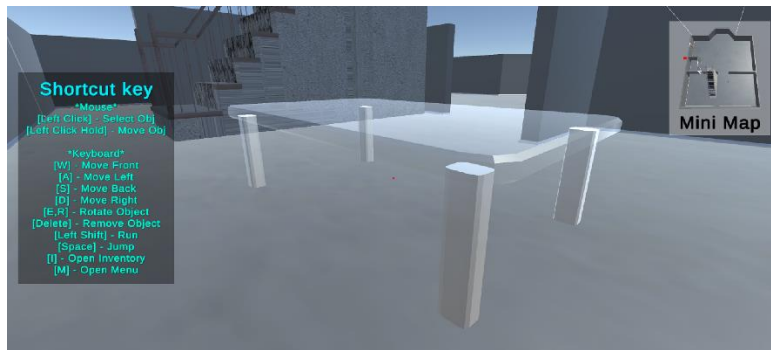
Screen Title : Double Storey Bed Room scene

Screen Layout	Description
  	<p>This scene shows the Double Storey Bed Room module of Room Design. When user entry this module a instruction will pop up and user can click “Next” button to skip it.</p> <p>User can control the character move around in this scene. There are some key used in this scene. Below are the function of these key:</p> <ul style="list-style-type: none"> • WASD – Character move • Left Shift – increase character move speed • Space – Character jump • M – open menu • I – open inventory • ER – Rotate object • Mouse left key click – select object • Mouse left key hold – Move object



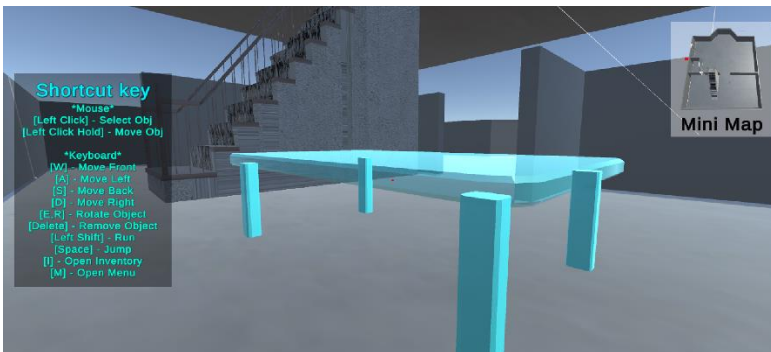
In Menu only have 4 button and 1 slider. Below are the functions of these items:

- Volume slider – to adjust the game volume
- “How to play” – to show the game instruction
- “view mode” – change the game mode
- “Back to main” – go to main menu

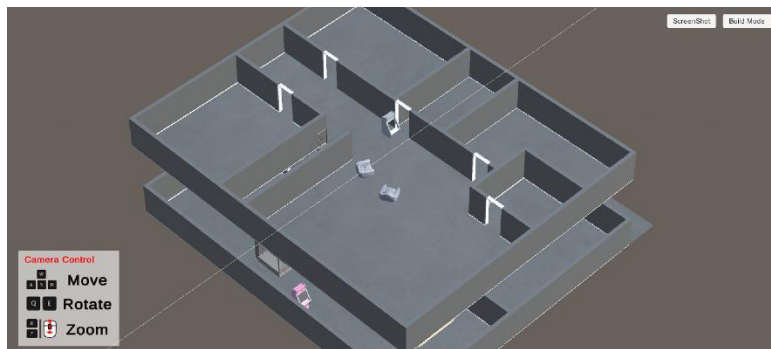


In Inventory there are some object button and 1 “x” button. Below are the function of the button:

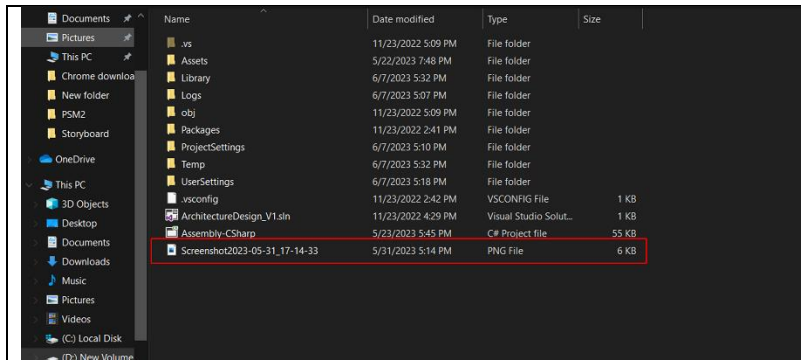
- Object button – spawn the object
- x - close the Inventory



Each object button will spawn different object. The object will spawn in front of the user and user can select the object to move, rotate and delete the object.

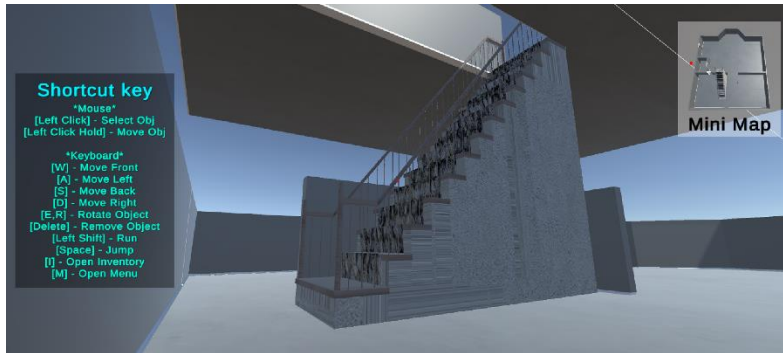


Once the user entry the view mode, there have some key



use in this mode. Below are the key function:

- WASD – Move the camera
- QE – Rotate the camera
- RF – Zoom in/zoom out camera



There are 2 button in view mode which is “Screenshot” and “build mode”. Screenshot can take the picture of the screen and save in the game file. Build mode for go back to default mode.

Table 3.7 Double Storey Bed Room Screen

3.7 SOFTWARE AND HARDWARE REQUIREMENT

Software and hardware is important to build and run the game application properly. Table 3.8.1 shows the software requirement that use for develop the game application and Table 3.8.2 shows the hardware requirement use for game application.

No	Software	Function
1	Unity	To build game space and game content
2	Microsoft Visual Studio 2019	To write script for game function
3	Adobe Photoshop	To edit the image

4	Draw.io	To draw the diagram
5	Canvas	To do the Gantt Chart
6	Google drive	To back up the file

Table 3.8 List of software requirement

No	Hardware	Function
1	Laptop/Desktop	Use to do the development and testing
2	Printer	To print the document

Table 3.9 List of hardware requirement

3.8 GANTT CHART

Refer to **APPENDIX A**

3.9 TESTING

User Acceptance Test (UAT) form is use to determine the functionality of the game application. Table 3.10 shows the UAT form that will be given to the user to fill it.

No	Event	Pass (✓) / Fail (✗)
Main Menu		
1	“Play” button can be press	
2	“Option” button can be press	
	- Full Screen can be select (Checkbox)	
	- Quality can be select (Multi option)	
	- Volume able to adjust (Slider)	
	- “←” button can be press	
3	“Exit” button can be press	
Choose Level		

1	“←” button can be press					
2	“1-Bed Room” Level can be select					
3	“2-Bed Room” Level can be select					
4	“3-Bed Room” Level can be select					
5	“Double Storey” Level can be select					
6	“Demo” Level can be select					
Level element		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	“Instruction” can be read and close					
2	“Inventory” can be open					
	- “Objects” button can be press					
	- “x” button can be press					
3	“Menu” can be open					
	- Volume able to adjust (Slider)					
	- “How to play” button can be press					
	- “View mode” button can be press					
	- “Back to main” button can be press					
	- “x” button can be press					
Character		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Character can move forward					
2	Character can move backward					
3	Character can move left					
4	Character can move right					
5	Character can run					
6	Character can jump					
7	Character can look around					
Object		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	Object can be spawn					

2	Object can be select				
3	Object can be move				
4	Object can be rotate				
5	Object can be delete				
6	Object can be deselect				
View Mode		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey
1	“Screenshot” button can be press				
2	Camera can be move				
3	Camera can be rotate				
4	Camera can be zoom				
5	“Build mode” button can be press				
Demo Level					
1	Menu can be open				
	- “Back to menu” button can be press				
	- “x” button can be press				

Table 3.10 User Acceptance Test (UAT) form

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter is about the development, implementation, and testing of Architecture game. This game is develop for student age 7 to 18 years. Software that use to develop this game application are Unity, Visual Studio 2019, and Adobe Photoshop. Testing is to identify the problem and error when run the application and fix the problem and error immediately.

4.2 DEVELOPMENT TOOLS

No	Tool	Purpose
1	Unity	To develop the game environment, element and gameplay
2	Visual Studio 2019	To write the code for the game
3	Adobe Photoshop	To edit the image

Table 4.1 Development tools for game application

4.3 IMPLEMENTATION

Implementation is a process that will record all the step of development Room Design game application.

4.3.1 Main Menu Scene

Main Menu consists a title text and 3 button, which is “PLAY”, “OPTION” and “EXIT”. These 3 button is put in a panel. The “PLAY” button is to navigate user to level scene. The “OPTION” button is to control the game volume and the last “EXIT” button is to allow user to close the game application. The image is use for the background of the Main Menu. Figure 4.1 shows the Main Menu scene of the game application.



Figure 4.1 Main Menu Scene

In Level scene consists 4 button which the user can choose the level to play, each button will bring user to different scene. Figure 4.2 shows the Level scene of the game application.



Figure 4.2 Level Scene

4.3.2 Level Scene

Figure 4.3 shows the level game scene. Once the user entry the game an instruction will shows in the middle of the screen. Figure 4.7 shows the instruction of the gameplay



Figure 4.3 Instruction of the gameplay

In the Game scene there are a game environment, a red dot in the middle of the scene and also a panel that show the shortcut key that user use to play the game. User can walk around in this environment and rotate the scene around. Red dot is use to detect the game object in the game and it also as a mouse position of the user. Figure 4.4 shows the game scene of the game.

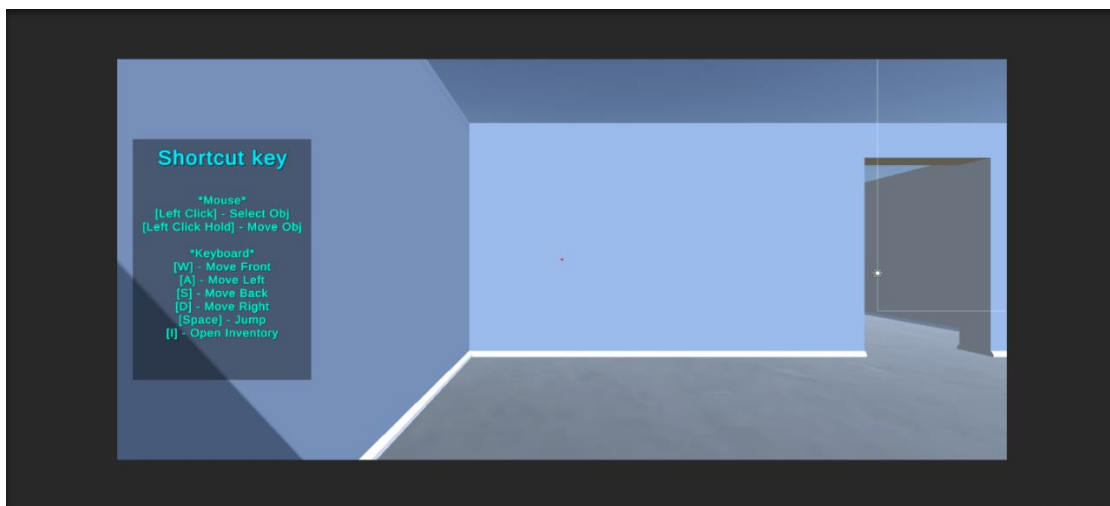


Figure 4.4 Game Scene

Figure 4.5 shows the Inventory scene that user can select the object and spawn it to the scene. In Inventory scene it have many button where the user can click it, at the top right side have a “X” button to close the inventory. Each of the white button refer to a different object to spawn. Once the object is spawn user can use the red dot locate to the object and the object will turn to transparent it means the object is detected. When the object is detected user can click “Mouse left button” to select the object, once the object has been selected the object will turn to green color. When the user hold “Mouse left button” and move around in the environment the object will also move based on the position of red dot. To release the object user just release the “Mouse left button”. If the user click “Mouse left button” without locate to any object in the scene it means the user unselect the object that select before and the color of the object will turn to original color. Figure 4.6, 4.7 and 4.8 show the object detect, object move and object deselect.

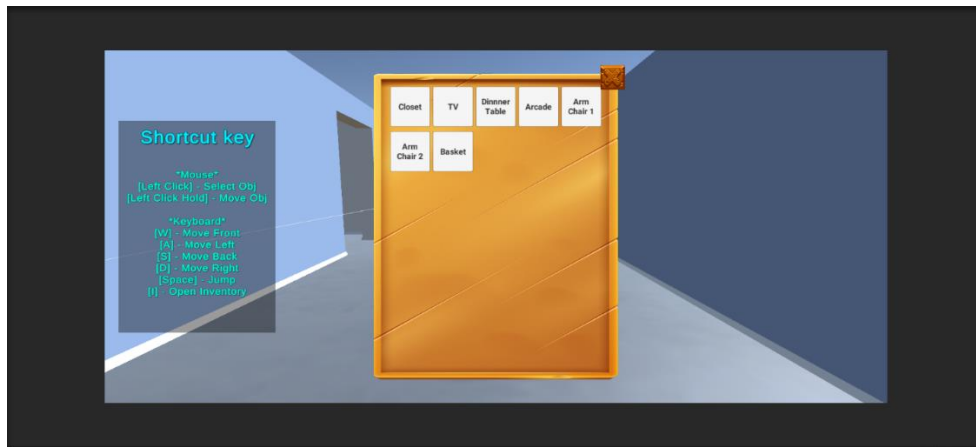


Figure 4.5 Inventory Scene

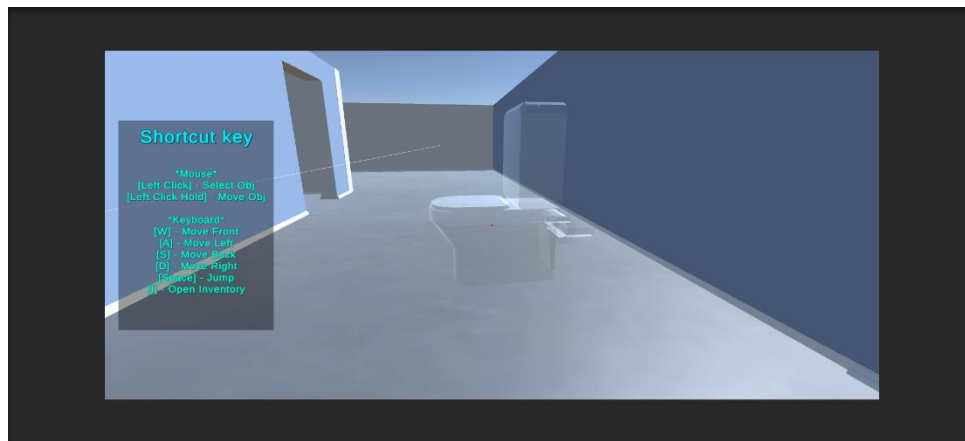


Figure 4.6 Object detected

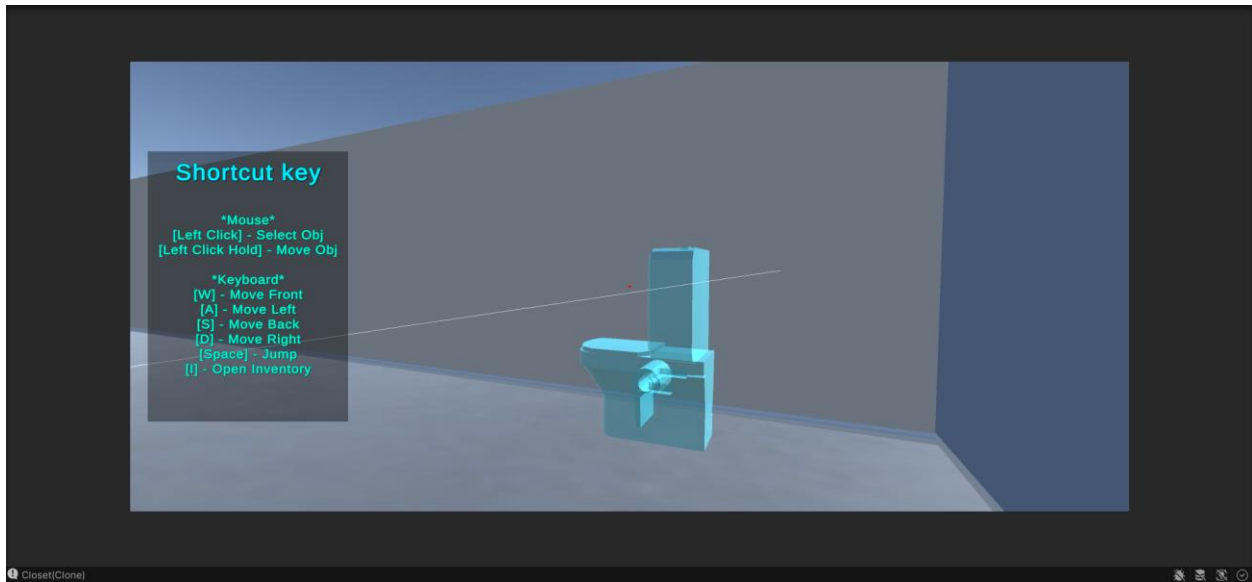


Figure 4.7 Object moving

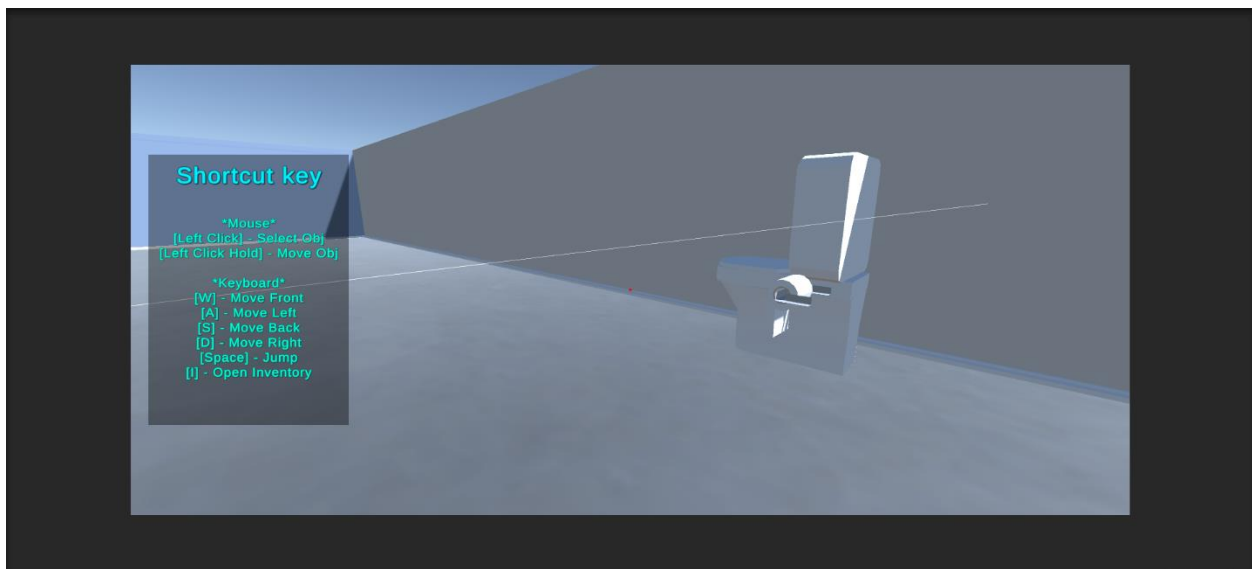


Figure 4.8 object deselected

Figure 4.9 shows the Menu of the game. In the menu user can adjust the game volume, read the game instruction, go to view mode and back to main menu. If user want to close the menu they can click on the “x” button on the top left side of the menu.

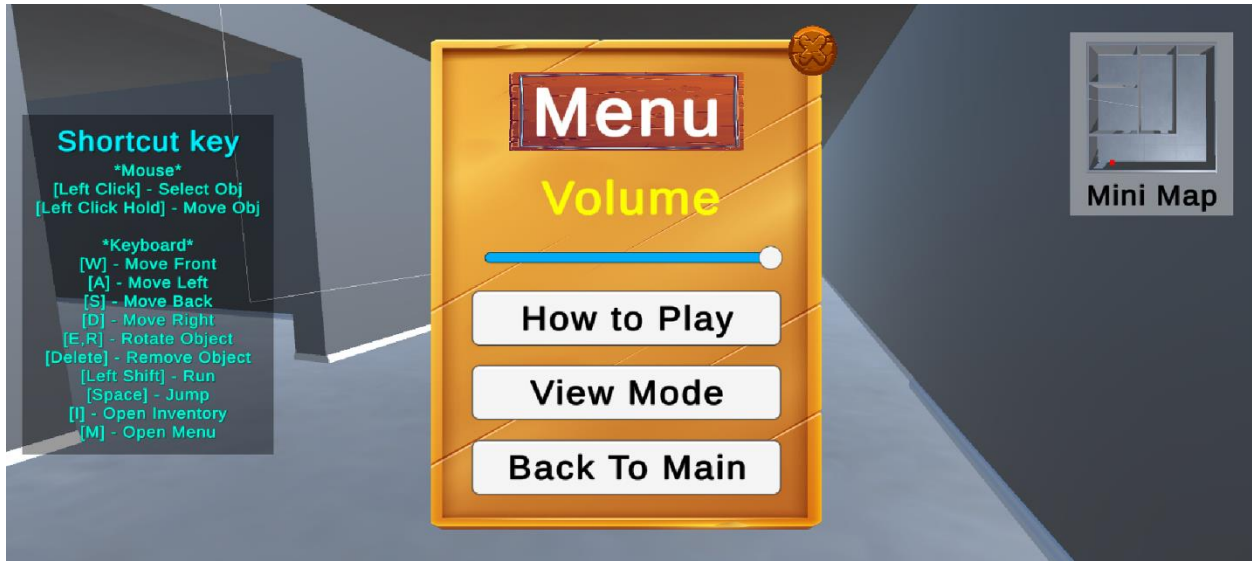


Figure 4.9 Game menu

Figure 4.10 shows the UI of the view mode. In view mode user can control their camera to view their design. User also can rotate and zoom the camera. In view mode user can click on “Screenshot” button to take a screenshot of their design and the picture will be saved in the game folder. If the user want to edit their design again they can click on the “build mode” to change the mode. Figure 4.11 shows the picture saved location.

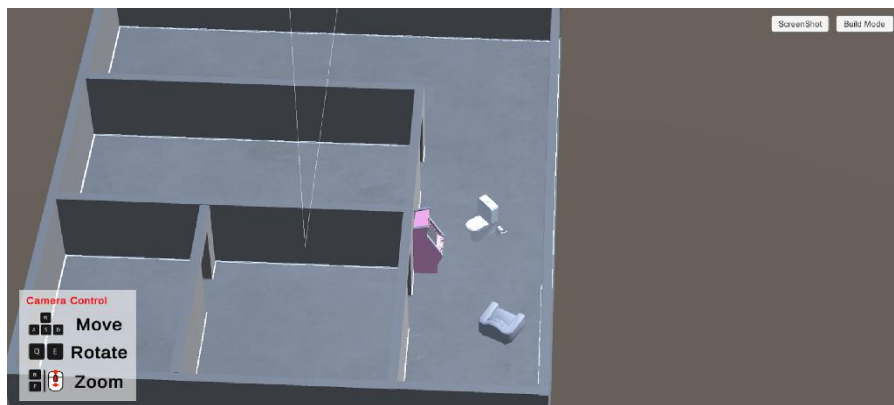


Figure 4.10 the UI of view mode

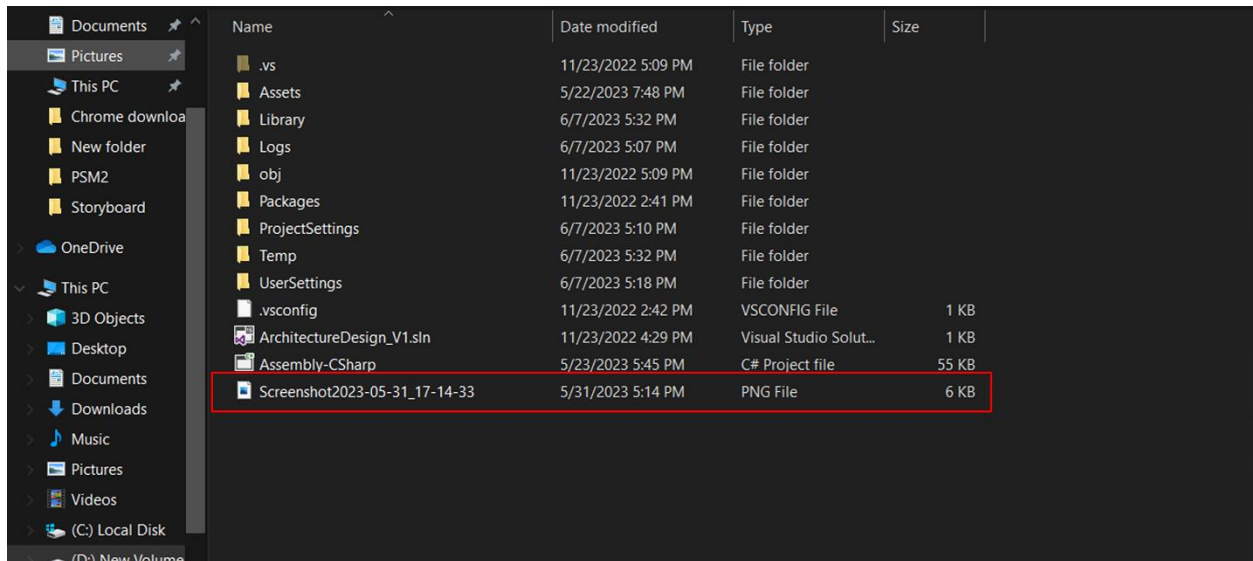


Figure 4.11 the picture saved location

4.4 CODING

Refer to **APPENDIX D**.

4.5 RESULT AND DISCUSSION

After game development, testing process will be done to evaluate the functionality and usability of the game application to make sure the game performance is good. To test the game, User Acceptance Test (UAT) is used to carry out where the user need to test the feature and function. User only need to use computer to run the UAT test and all the result will be recorded. Refer to **APPENDIX C** for the result of UAT.

To test the usability and effectiveness of the game application, a usability survey form using google form (refer to **APEENDIX B**) was given to the users to fill the form. There are some multiple choose question and fillable question that user need to fill it. Figure 4.12 shows the result of the Game UI survey and Figure 4.13 shows the result of game experience.

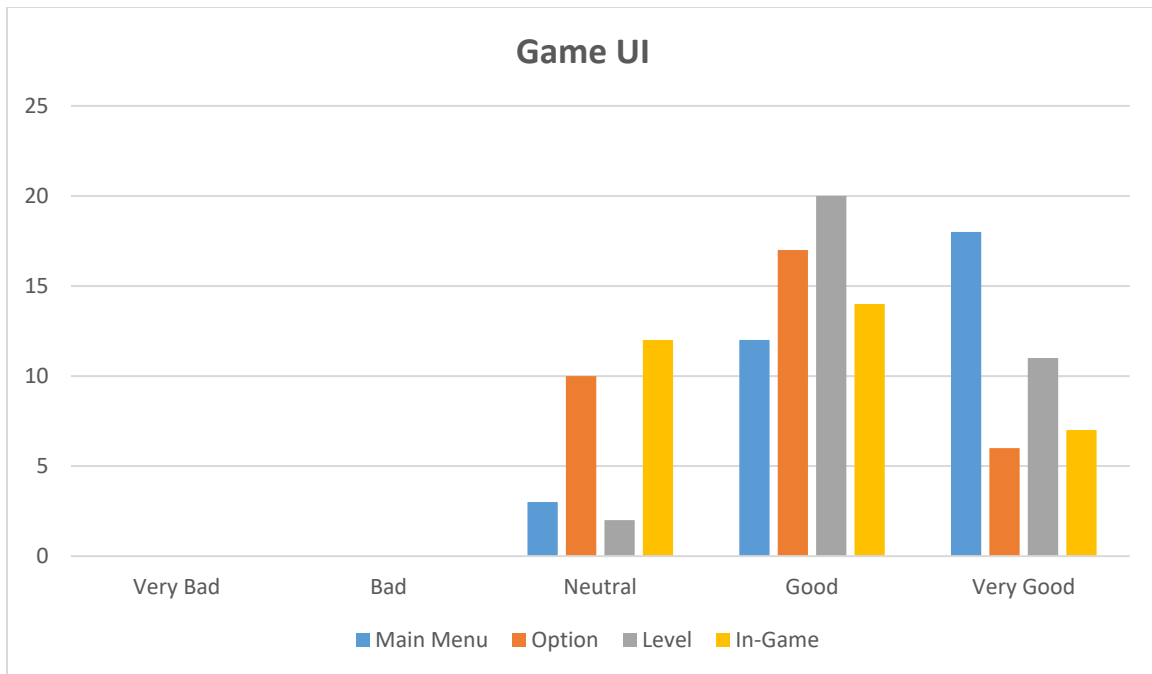


Figure 4.12 survey on game UI

Based on the graph above, 54.5% of the user think the main menu UI of the game application is very good, 36.4% of the user feel good to the design and only 9.1% user feel neutral to the design. For the UI design of option there are 69.7% of user like the design and 30.3% feel neutral to the design. For level UI design there are 93.9% of the user like the design and only 6.1% user feel neutral. Lastly, for In-Game UI design there are 63.6% of the user like the design and 36.4% feel neutral to the design. From the result, the design of the Main menu, Option and Level are nice to the user and for the In-game UI need to be improve.



Figure 4.13 survey of game experience to the user

Based on the graph above, 97% of the user easy to know how to play the game and 3% of the user feel neutral to the game difficulty to play the game. Other that, 93.9% of the user feel the game is very smooth and 6.1% of the user feel neutral to the smoothness of the game. There are 69.7% of the user understand the game instruction given and 30.3% of the user think that the game instruction are not clear enough. 81.8% of the user didn't have any bug or glitches when playing the game and 18.2% of the user have bug or glitches when playing the game. 97% of the user think the bug or glitches didn't affect the gameplay and only 3% of the user think the bug or glitches affect the gameplay. Therefore, based on the survey it can be concluded that the game application received positive feedback from the users but it also need to be improve on the gameplay.

Lastly, there are some comments given by users to improve the game application. These comments are summarized in Table 4.2.

No	Comments
1	Graphic need to improve

2	More function to the game, example resize the object, editable game space
3	The design of UI and need to improve
4	The object moving not so smooth

Table 4.2 Summary of user's comments

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

This chapter will discuss the summary of the game to achieve the objectives and overcome the problems. Most of the time when we want to design our house interior we will find a designer or make the design on the professional design application such as AutoCAD & SketchUp. It will cost a lot of money to do it. This game application can serve as a supported tool to simulate the design that user want because it contains the object and function for design. User can easily to make their own design through this game application. This game application is made by Unity, Microsoft Visual studio and Adobe Photoshop as additional software to develop.

5.2 LIMITATION AND CONSTRAINT

i. Time

The development time is limited so the game application must be complete in time. Due to the time limited so the game element and module also will be limited. All the game element is downloaded from internet and only a few module will be developed.

ii. Coding

Coding is most important part to add the function to a game. When doing the coding there are some error(s) occur in the script. This error(s) will made the game application unable to run until the error(s) solve so it will take time to find the solution for the error(s)

iii. Game function

There only some basic game function in this game application to make the game playable. For additional function such as save and load didn't implement in the game application due the limitation of the skill.

5.3 FUTURE WORK

There are some features that can be added or improve for this game application.

i. Developer can replace the object to high reality object (Materials)

ii. Developer can make the map become editable. (Example: can add/delete map space based on their need.

iii. Developer can add more type of module in game application (Example: Garden, Office, School, etc.)

iv. Developer can build the game on more platform (Example: Mobile app and Web application)

v. Developer can add more language into the game application (Example: Chinese, Bahasa Melayu, etc.)

REFERENCE

- Wikipedia (2021). Game classification. https://en.wikipedia.org/wiki/Game_classification
- House Flipper official (2021). Game website. <https://www.houseflippergame.com/>
- Wawro, Alex (29 May 2018). "Teachers can now access an educational version of Cities: Skylines" <https://www.gamedeveloper.com/business/teachers-can-now-access-an-educational-version-of-i-cities-skylines-i->
- Steam (2015). Game Store. https://store.steampowered.com/app/255710/Cities_Skylines/
- Wikipedia (2014). https://en.wikipedia.org/wiki/The_Sims_4
- Steam (2014). Game Store. https://store.steampowered.com/app/1222670/The_Sims_4/
- The Sims 4 official (2014). Game website. <https://www.ea.com/zh-cn/games/the-sims/the-sims-4>
- Starloop Studios (2020). GAME TESTING 101: BASIC TIPS AND STRATEGIES. Spain. <https://starloopstudios.com/game-testing-101-tips-and-strategies/>
- Klabbers, J.H.G. (1999). Three easy pieces: A taxonomy of gaming. In D. Saunders, & J. Severn (Eds.), The international simulation and gaming research yearbook. Vol. 7: Simulations and games for strategy and policy planning (pp. 16-33). London: Kogan Page Limited. <https://scholar.google.com/scholar?q=Klabbers>
- Elsevier Ltd (2012). Testing a simulation game as a potential teaching method for a masters course in human resources management. <https://www.sciencedirect.com/science/article/pii/S1877042812002492>
- SDLC - Agile Model. (n.d.). www.tutorialspoint.com. https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm#:~:text=What%20is%20Agile%3F

APPENDIX A

TASK	DURATION	QUARTER 1				QUARTER 2				QUARTER 3			
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Planning													
Identify objective	7d			■									
Identify project scope	7d			■									
Project research	25d			■									
Design													
Storyboard design	10d					■							
Multimedia element design	15d						■						
Develop													
Game content	75d							■					
Game function	75d							■					
Build game	3d										■		
Test													
User Acceptance Test	7d										■		
Deploy													
Debug	30d										■		
Review													
Usability Test	30d											■	
Launch													
Game upload	3d												■

Figure 1 Gantt chart of the project

APPENDIX B

USABILITY TEST (QUESTION FOR USER)

The figure displays two screenshots of a Google Form titled "Usability Testing for Architecture Game".

Left Screenshot (Introduction):

- Header: "Usability Testing for Architecture Game"
- Text: "Hello everyone, I'm Tan Shao Kang from Faculty of Computing in University Malaysia Pahang. Here is my final year project and the purpose for doing this survey is to get the feedback from the user and improve the game. This game is PC game so please make sure you're using PC to doing this survey. Here is the link to download my project : [Project](#)"
- Text: "Please take a few minutes to fill up this survey, Thank you."
- Footer: "shadowking981229@gmail.com", "Switch account", and a privacy notice: "The name and photo associated with your Google account will be recorded when you upload files and submit this form. Your email is not part of your response."
- Progress bar: "Page 1 of 5"
- Buttons: "Next" and "Clear" (with an edit icon).

Right Screenshot (Section 1: Personal Information):

- Section Header: "Section 1 : Personal Information"
- Question 1: "What is your age? *"
- Input: A dropdown menu showing "8-12".
- Question 2: "What is your gender? *"
- Options: Radio buttons for "Male" (selected) and "Female".
- Buttons: "Next" and "Clear" (with an edit icon).

Figure 2 Usability test question using Google Form (Section 1)

The figure displays two screenshots of a Google Form titled "Section 2 : Game UI". The form contains four usability test questions, each with a 5-point Likert scale ranging from "Very Bad" (1) to "Very Good" (5). The first screenshot shows the "Main Menu Interface" and "Option Interface" questions. The second screenshot shows the "Level Interface" and "In-Game Interface" questions. In the second screenshot, the "Level Interface" question has a rating of 4 selected, and the "In-Game Interface" question has a rating of 4 selected. The form also includes a progress indicator at the bottom right of the second screenshot, showing "Page 3 of 5".

Section 2 : Game UI

Main Menu Interface *

Very Bad

1

2

3

4

5

Very Good

Option Interface *

Very Bad

1

2

3

4

5

Very Good

Level Interface *

Very Bad

1

2

3

4

5

Very Good

In-Game Interface *

Very Bad

1

2

3

4

5

Very Good

Page 3 of 5

Figure 3 Usability test question using Google Form (Section 2)

22:15 99%

docs.google.com/for

Section 3 : Game Functionability

Can the game application run well? *

Yes
 No

All the **button** can function well *

Yes
 No

If no, please screenshot the functionless button and upload it

[Add file](#)

Character is controllable *

Yes
 No

22:15 99%

If no, please screenshot the functionless button and upload it

[Add file](#)

Character is controllable *

Yes
 No

Object can be spawn, control and delete *

Yes
 No

Page 4 of 5

[Back](#) [Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google.
[Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 4 Usability test question using Google Form (Section 3)

22:15 99%

Section 4 : Game Experience

Game is easy to play *

Strongly disagree

1

2

3

4

5

Strongly agree

Smoothness of the game *

very Smoothless

1

2

3

4

5

very Smooth

22:15 99%

All the instructions given are clear and easy to understand *

Yes

No

Does the game have a lot of bugs and glitches? *

Many

A few

No

Does the bugs and glitches effect the gameplay? *

Yes

No

Suggestion and feature for the game

Your answer

Figure 5 Usability test question using Google Form (Section 4)

22:15 99%

No

Does the bugs and glitches effect the gameplay? *

Yes

No

Suggestion and feature for the game

Your answer

That's all for survey, Thanks for your response. your response will be recorded for improve the game to become more better. Untitled Title

Page 5 of 5

Back Submit Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google.
[Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 6 Usability test question using Google Form (Section 4)

APPENDIX C

USER ACCEPTANCE TEST (UAT)

User Acceptance Test (UAT)

No: 1 Name: Tan Yuan Jue

No	Event	Pass (✓) / Fail (✗)			
Main Menu					
1	"Play" button can be press	/			
2	"Option" button can be press	/			
	- Full Screen can be select (Checkbox)	/			
	- Quality can be select (Multi option)	/			
	- Volume able to adjust (Slider)	/			
	- "←" button can be press	/			
3	"Exit" button can be press	/			
Choose Level					
1	"←" button can be press	/			
2	"1-Bed Room" Level can be select	/			
3	"2-Bed Room" Level can be select	/			
4	"3-Bed Room" Level can be select	/			
5	"Double Storey" Level can be select	/			
6	"Demo" Level can be select	/			
Level element		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey
1	"Instruction" can be read and close	/	/	/	/
2	"Inventory" can be open	/	/	/	/
	- "Objects" button can be press	/	/	/	/
	- "x" button can be press	/	/	/	/
3	"Menu" can be open	/	/	/	/
	- Volume able to adjust (Slider)	/	/	/	/
	- "How to play" button can be press	/	/	/	/

Figure 7 UAT Form result for 1st person

	- "View mode" button can be press	/	/	/	/	
	- "Back to main" button can be press	/	/	/	/	
	- "x" button can be press	/	/	/	/	
Character		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Character can move forward	/	/	/	/	/
2	Character can move backward	/	/	/	/	/
3	Character can move left	/	/	/	/	/
4	Character can move right	/	/	/	/	/
5	Character can run	/	/	/	/	/
6	Character can jump	/	/	/	/	/
7	Character can look around	/	/	/	/	/
Object		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	Object can be spawn	/	/	/	/	/
2	Object can be select	/	/	/	/	/
3	Object can be move	/	/	/	/	/
4	Object can be rotate	/	/	/	/	/
5	Object can be delete	/	/	/	/	/
6	Object can be deselect	/	/	/	/	/
View Mode		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	"Screenshot" button can be press	/	λ	×	×	
2	Camera can be move	/	/	/	/	
3	Camera can be rotate	/	/	/	/	
4	Camera can be zoom	/	/	/	/	
5	"Build mode" button can be press	/	/	/	/	
Demo Level						
1	Menu can be open	/	/	/	/	/
	- "Back to menu" button can be press	/	/	/	/	/

Figure 8 UAT Form result for 1st person

- "x" button can be press	/	/	/	/
---------------------------	---	---	---	---

Signature:

Jue

Figure 9 UAT Form result for 1st person

User Acceptance Test (UAT)

No: 2

Name: *Chung Men Hong*

No	Event	Pass (✓) / Fail (✗)			
Main Menu					
1	"Play" button can be press	/			
2	"Option" button can be press	/			
	- Full Screen can be select (Checkbox)	/			
	- Quality can be select (Multi option)	/			
	- Volume able to adjust (Slider)	/			
	- "←" button can be press	/			
3	"Exit" button can be press	/			
Choose Level					
1	"←" button can be press	/			
2	"1-Bed Room" Level can be select	/			
3	"2-Bed Room" Level can be select	/			
4	"3-Bed Room" Level can be select	/			
5	"Double Storey" Level can be select	/			
6	"Demo" Level can be select	/			
Level element		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey
1	"Instruction" can be read and close	/	/	/	/
2	"Inventory" can be open	/	/	/	/
	- "Objects" button can be press	/	/	/	/
	- "x" button can be press	/	/	/	/
3	"Menu" can be open	/	/	/	/
	- Volume able to adjust (Slider)	/	/	/	/
	- "How to play" button can be press	/	/	/	/

Figure 10 UAT Form result for 2nd person

	- "View mode" button can be press	/	/	/	/	
	- "Back to main" button can be press	/	/	/	/	
	- "x" button can be press	/	/	/	/	
Character		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Character can move forward	/	/	/	/	/
2	Character can move backward	/	/	/	/	/
3	Character can move left	/	/	/	/	/
4	Character can move right	/	/	/	/	/
5	Character can run	/	/	/	/	/
6	Character can jump	/	/	/	/	/
7	Character can look around	/	/	/	/	/
Object		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	Object can be spawn	/	/	/	/	/
2	Object can be select	/	/	/	/	/
3	Object can be move	/	/	/	/	/
4	Object can be rotate	/	/	/	/	/
5	Object can be delete	/	/	/	/	/
6	Object can be deselect	/	/	/	/	/
View Mode		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	
1	"Screenshot" button can be press	/	/	/	/	/
2	Camera can be move	/	/	/	/	/
3	Camera can be rotate	/	/	/	/	/
4	Camera can be zoom	/	/	/	/	/
5	"Build mode" button can be press	/	/	/	/	/
Demo Level						
1	Menu can be open	/	/	/	/	/
	- "Back to menu" button can be press	/	/	/	/	/

Figure 11 UAT Form result for 2nd person

- "x" button can be press	/	/	/	/
---------------------------	---	---	---	---

Signature:

Chong

Figure 12 UAT Form result for 2nd person

User Acceptance Test (UAT)

No: 3

Name: CHENG YUN JUN

No	Event	Pass (✓) / Fail (✗)			
Main Menu					
1	“Play” button can be press	✓			
2	“Option” button can be press	✓			
	- Full Screen can be select (Checkbox)	✓			
	- Quality can be select (Multi option)	✓			
	- Volume able to adjust (Slider)	✓			
	- “←” button can be press	✓			
3	“Exit” button can be press	✓			
Choose Level					
1	“←” button can be press	✓			
2	“1-Bed Room” Level can be select	✓			
3	“2-Bed Room” Level can be select	✓			
4	“3-Bed Room” Level can be select	✓			
5	“Double Storey” Level can be select	✓			
6	“Demo” Level can be select	✓			
Level element		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey
1	“Instruction” can be read and close	✓	✓	✓	✓
2	“Inventory” can be open	✓	✓	✓	✓
	- “Objects” button can be press	✓	✓	✓	✓
	- “x” button can be press	✓	✓	✓	✓
3	“Menu” can be open	✓	✓	✓	✓
	- Volume able to adjust (Slider)	✓	✓	✓	✓
	- “How to play” button can be press	✓	✓	✓	✓

Figure 13 UAT Form result for 3rd person

	- "View mode" button can be press	✓	✓	✓	✓	
	- "Back to main" button can be press	✓	✓	✓	✓	
	- "x" button can be press	✓	✓	✓	✓	
Character		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Character can move forward	✓	✓	✓	✓	✓
2	Character can move backward	✓	✓	✓	✓	✓
3	Character can move left	✓	✓	✓	✓	✓
4	Character can move right	✓	✓	✓	✓	✓
5	Character can run	✓	✓	✓	✓	✓
6	Character can jump	✓	✓	✓	✓	✓
7	Character can look around	✓	✓	✓	✓	✓
Object		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Object can be spawn	✓	✓	✓	✓	✓
2	Object can be select	✓	✓	✓	✓	✓
3	Object can be move	✓	✓	✓	✓	✓
4	Object can be rotate	✓	✓	✓	✓	✓
5	Object can be delete	✓	✓	✓	✓	✓
6	Object can be deselect	✓	✓	✓	✓	✓
View Mode		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	"Screenshot" button can be press	✓	✓	✓	✓	✓
2	Camera can be move	✓	✓	✓	✓	✓
3	Camera can be rotate	✓	✓	✓	✓	✓
4	Camera can be zoom	✓	✓	✓	✓	✓
5	"Build mode" button can be press	✓	✓	✓	✓	✓
Demo Level		1-Bed Room	2-Bed Room	3-Bed Room	Double Storey	Demo
1	Menu can be open	✓	✓	✓	✓	✓
	- "Back to menu" button can be press	✓	✓	✓	✓	✓

Figure 14 UAT Form result for 3rd person

-	"x"	button	can	be	press	✓	✓	✓	✓
---	-----	--------	-----	----	-------	---	---	---	---

Signature:

cheng

Figure 15 UAT Form result for 3rd person

APPENDIX D

CODING

```
public void PlayGame()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
}
```

Figure 16 Script for play the game

```
public void Option()
{
    SceneManager.LoadScene("Option");
}
```

Figure 17 Script for open the option

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

public class Option : MonoBehaviour
{
    public AudioManager mainMixer;
    public void SetVolume(float volume)
    {
        mainMixer.SetFloat("volume", volume);
    }

    public void SetFullScreen(bool isFullscreen)
    {
        Screen.fullScreen = isFullscreen;
    }

    public void SetQuality(int qualityIndex)
    {
        QualitySettings.SetQualityLevel(qualityIndex);
    }
}
```

Figure 18 Script for Option

```

public void QuitGame()
{
    //Debug.Log("QUIT!");
    Application.Quit();
}

```

Figure 19 Script for quit the game

```

public void Level1()
{
    //SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    SceneManager.LoadScene("1BedRoom");
}

public void Level2()
{
    SceneManager.LoadScene("2BedRoom");
}

public void Level3()
{
    SceneManager.LoadScene("3BedRoom");
}

public void Level4()
{
    SceneManager.LoadScene("DoubleStoreyHouse");
}

public void Demo()
{
    SceneManager.LoadScene("Demo");
}

```

Figure 20 Script for load the module

```

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

public class InsStart : MonoBehaviour
{
    public GameObject buildCanvas;
    public GameObject instructionCanvas;
    public GameObject firstCanvas;
    public GameObject secondCanvas;
    public GameObject thirdCanvas;
    public Behaviour playerScript;
    public GameObject Menu;

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

```

```

    {
        openInstruction();
    }

    public void openInstruction()
    {
        instructionCanvas.SetActive(true);
        buildCanvas.SetActive(false);
        firstCanvas.SetActive(true);
        secondCanvas.SetActive(false);
        thirdCanvas.SetActive(false);
        Cursor.lockState = CursorLockMode.None;
        Cursor.visible = true;
        playerScript.enabled = false;
    }

    public void secondInstruction()
    {
        firstCanvas.SetActive(false);
        secondCanvas.SetActive(true);
    }

    public void thirdInstruction()
    {
        secondCanvas.SetActive(false);
        thirdCanvas.SetActive(true);
    }

    public void closeInstruction()
    {
        buildCanvas.SetActive(true);
        thirdCanvas.SetActive(false);
        instructionCanvas.SetActive(false);
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
        playerScript.enabled = true;
        Menu.SetActive(false);
    }
}

```

Figure 21 Script for display the game instruction when in-game

```

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

public class Player : MonoBehaviour
{
    public int moveSpeed;
    public bool middleMouse;
    [SerializeField] Transform playerCamera;
    [SerializeField] [Range(0.0f, 0.5f)] float mouseSmoothTime = 0.03f;
    [SerializeField] bool cursorLock = true;
    [SerializeField] float mouseSensitivity = 3.5f;
}

```

```

[SerializeField] float speed = 6.0f;
[SerializeField] [Range(0.0f, 0.5f)] float moveSmoothTime = 0.3f;
[SerializeField] float gravity = -30f;
[SerializeField] Transform groundCheck;
[SerializeField] LayerMask ground;
[SerializeField] float jumpHeight = 6f;
CharacterController controller;

private float velocityY;
private bool isGrounded;
private float cameraCap;

Vector2 currentMouseDelta;
Vector2 currentMouseDeltaVelocity;
Vector2 currentDir;
Vector2 currentDirVelocity;
Vector2 velocity;

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

    if (cursorLock)
    {
        Cursor.lockState = CursorLockMode.Locked; //Cursor Locked
        Cursor.visible = false; //Cursor is invisible
        middleMouse = true;
    }
}

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

private void UpdateMove()
{
    Vector2 targetMouseDelta = new Vector2(Input.GetAxis("Mouse X"),
Input.GetAxis("Mouse Y"));

    currentMouseDelta = Vector2.SmoothDamp(currentMouseDelta, targetMouseDelta, ref
currentMouseDeltaVelocity, moveSmoothTime);

    cameraCap -= currentMouseDelta.y * mouseSensitivity;

    cameraCap = Mathf.Clamp(cameraCap, -90.0f, 90.0f);

    playerCamera.localEulerAngles = Vector3.right * cameraCap;

    transform.Rotate(Vector3.up * currentMouseDelta.x * mouseSensitivity);
}

private void UpdateMouse()
{

```

```

        isGrounded = Physics.CheckSphere(groundCheck.position, 0.2f, ground);

        Vector2 targetDir = new Vector2(Input.GetAxisRaw("Horizontal"),
Input.GetAxisRaw("Vertical"));
        targetDir.Normalize();

        currentDir = Vector2.SmoothDamp(currentDir, targetDir, ref currentDirVelocity,
mouseSmoothTime);

        velocityY += gravity * 2f * Time.deltaTime;

        Vector3 velocity = (transform.forward * currentDir.y + transform.right *
currentDir.x) * speed + Vector3.up * velocityY;

        controller.Move(velocity * Time.deltaTime);

        if (isGrounded && Input.GetButtonDown("Jump"))
        {
            velocityY = Mathf.Sqrt(jumpHeight * -2f * gravity);
        }

        if(isGrounded! && controller.velocity.y < -1f)
        {
            velocityY = -8f;
        }
    }
}

```

Figure 22 Script for middle the mouse position to the screen and player control

```

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

public class InventoryManager : MonoBehaviour
{
    public GameObject buildCanvas;
    public GameObject viewCanvas;
    public GameObject player;
    public GameObject viewCam;
    public GameObject inventory;
    public Behaviour playerScript;
    public GameObject menu;
    public Behaviour selectScript;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.I))
        {
            inventory.SetActive(true);
            menu.SetActive(false);
            Cursor.lockState = CursorLockMode.None;
            Cursor.visible = true;
        }
    }
}

```



```

        playerScript.enabled = false;
    }

    if (Input.GetKey(KeyCode.M))
    {
        menu.SetActive(true);
        inventory.SetActive(false);
        Cursor.lockState = CursorLockMode.None;
        Cursor.visible = true;
        playerScript.enabled = false;
    }
}

public void CloseInventory(GameObject inventory)
{
    inventory.SetActive(false);
    Cursor.lockState = CursorLockMode.Locked;
    Cursor.visible = false;
    playerScript.enabled = true;
}

public void CloseMenu(GameObject menu)
{
    menu.SetActive(false);
    Cursor.lockState = CursorLockMode.Locked;
    Cursor.visible = false;
    playerScript.enabled = true;
}

public void ToViewMode()
{
    buildCanvas.SetActive(false);
    viewCanvas.SetActive(true);
    player.SetActive(false);
    viewCam.SetActive(true);
    selectScript.enabled = false;
}

public void ToBuildMode()
{
    viewCanvas.SetActive(false);
    buildCanvas.SetActive(true);
    player.SetActive(true);
    viewCam.SetActive(false);
    menu.SetActive(false);
    selectScript.enabled = true;
    playerScript.enabled = true;
    Cursor.lockState = CursorLockMode.Locked;
    Cursor.visible = false;
}
}

```

Figure 23 Script for open the Inventory and Menu UI

```

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

public class Spawner : MonoBehaviour
{
    public GameObject[] objects;
    private GameObject objectSpawn;

    // Update is called once per frame

    public void SpawnObject(int index)
    {
        objectSpawn = Instantiate(objects[index], transform.position,
Quaternion.identity);
    }
}

```

Figure 24 Script for spawn the object

```

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

public class Selection : MonoBehaviour
{
    public Material[] highlightMaterial = new Material[5];
    public Material[] selectionMaterial = new Material[5];

    private Material[] originalMaterial = new Material[5];
    private Transform highlight;
    private Transform selection;
    private RaycastHit raycastHit;

    public Transform holdPos;
    //private float rotationSensitivity = 1f; //how fast/slow the object is rotated in
relation to mouse movement
    public float turnspeed;

    private void FixedUpdate()
    {
    }

    // Update is called once per frame
    void Update()
    {
        //Highlight
        if (highlight != null)
        {
            highlight.GetComponent<MeshRenderer>().materials = originalMaterial;
            highlight = null;
        }
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    }
}

```

```

    if (!EventSystem.current.IsPointerOverGameObject() && Physics.Raycast(ray, out
raycastHit)) //Make sure you have EventSystem in the hierarchy before using EventSystem
    {
        highlight = raycastHit.transform;
        if(highlight.CompareTag("Selectable") && highlight != selection)
        {
            if(highlight.GetComponent<MeshRenderer>().materials !=
highlightMaterial)
            {
                originalMaterial =
highlight.GetComponent<MeshRenderer>().materials;
                highlight.GetComponent<MeshRenderer>().materials =
highlightMaterial;
            }
            else
            {
                highlight = null;
            }
        }

        //Selection
        if (Input.GetKey(KeyCode.Mouse0)
&& !EventSystem.current.IsPointerOverGameObject()) //Make sure you have EventSystem in
the hierarchy before using EventSystem
        {

            if (selection != null)
            {
                selection.GetComponent<MeshRenderer>().materials = originalMaterial;
                selection = null;
            }

            if (!EventSystem.current.IsPointerOverGameObject() && Physics.Raycast(ray,
out raycastHit))
            {
                selection = raycastHit.transform;

                if (selection.CompareTag("Selectable"))
                {
                    selection.GetComponent<MeshRenderer>().materials =
selectionMaterial;
                    Debug.Log(selection.gameObject.name);
                    MoveObject();
                    if(Input.GetKey(KeyCode.R))
                    {
                        RotateObjectClockwise();
                    }
                    if(Input.GetKey(KeyCode.E))
                    {
                        RotateObjectAntiClockwise();
                    }
                    if (Input.GetKey(KeyCode.Delete))
                    {
                        DeleteObject();
                    }
                }
            }
            else

```

```

        {
            selection = null;
        }
    }
}

```

Figure 25 Script for select the object

```

void MoveObject()
{
    //holdPos.position = raycastHit.transform.position;
    selection.transform.position = holdPos.transform.position;
}

```

Figure 26 Script for drag and move the object

```

void RotateObjectClockwise()
{
    selection.transform.Rotate(Vector3.up * turnspeed * Time.deltaTime);
}

void RotateObjectAntiClockwise()
{
    selection.transform.Rotate(Vector3.up * turnspeed * -Time.deltaTime);
}

```

Figure 27 Script for rotate the object

```

void DeleteObject()
{
    Destroy(selection.gameObject);
}

```

Figure 28 Script for delete the object

```

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

public class CameraController : MonoBehaviour
{
    public Transform cameraTransform;

    public float normalSpeed;
    public float fastSpeed;
    public float movementSpeed;
    public float movementTime;
}

```

```

public float rotationAmount;

public Vector3 zoomAmount;
public Vector3 newZoom;
public Vector3 newPosition;
public Quaternion newRotation;

public Vector3 dragStartPosition;
public Vector3 dragCurrentPosition;
public Vector3 rotateStartPosition;
public Vector3 rotateCurrentPosition;

// Start is called before the first frame update
void Start()
{
    newPosition = transform.position;
    newRotation = transform.rotation;
    newZoom = cameraTransform.localPosition;
}

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

void HandleMouseInput()
{
    if(Input.mouseScrollDelta.y != 0)
    {
        newZoom += Input.mouseScrollDelta.y * zoomAmount;
    }

    if(Input.GetMouseButton(0))
    {
        Plane plane = new Plane(Vector3.up, Vector3.zero);
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        float entry;
        if(plane.Raycast(ray, out entry))
        {
            dragStartPosition = ray.GetPoint(entry);
        }
    }
    if(Input.GetMouseButton(0))
    {
        Plane plane = new Plane(Vector3.up, Vector3.zero);
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        float entry;
        if (plane.Raycast(ray, out entry))
        {
            dragCurrentPosition = ray.GetPoint(entry);
            newPosition = transform.position + dragStartPosition -
dragCurrentPosition;
        }
    }

    if(Input.GetMouseButtonDown(2))

```

```

    {
        rotateStartPosition = Input.mousePosition;
    }

    if(Input.GetMouseButton(2))
    {
        rotateCurrentPosition = Input.mousePosition;
        Vector3 difference = rotateStartPosition - rotateCurrentPosition;
        rotateStartPosition = rotateCurrentPosition;
        newRotation *= Quaternion.Euler(Vector3.up * (-difference.x / 5f));
    }
}

void HandleMovementInput()
{
    if(Input.GetKey(KeyCode.LeftShift))
    {
        movementSpeed = fastSpeed;
    }
    else
    {
        movementSpeed = normalSpeed;
    }

    if(Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.UpArrow))
    {
        newPosition += (transform.forward * movementSpeed);
    }

    if(Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow))
    {
        newPosition += (transform.forward * -movementSpeed);
    }

    if(Input.GetKey(KeyCode.D) || Input.GetKey(KeyCode.RightArrow))
    {
        newPosition += (transform.right * movementSpeed);
    }

    if(Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.LeftArrow))
    {
        newPosition += (transform.right * -movementSpeed);
    }

    if (Input.GetKey(KeyCode.Q))
    {
        newRotation *= Quaternion.Euler(Vector3.up * rotationAmount);
    }
    if (Input.GetKey(KeyCode.E))
    {
        newRotation *= Quaternion.Euler(Vector3.up * -rotationAmount);
    }

    if(Input.GetKey(KeyCode.R))
    {
        newZoom += zoomAmount;
    }
    if(Input.GetKey(KeyCode.F))

```

```

    {
        newZoom -= zoomAmount;
    }

    transform.position = Vector3.Lerp(transform.position, newPosition,
Time.deltaTime * movementTime);
    transform.rotation = Quaternion.Lerp(transform.rotation, newRotation,
Time.deltaTime * movementTime);
    cameraTransform.localPosition = Vector3.Lerp(cameraTransform.localPosition,
newZoom, Time.deltaTime * movementTime);
    }
}

```

Figure 29 Script for move, rotate and zoom the camera in view mode

```

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

public class ScreenShot : MonoBehaviour
{
    public GameObject canvas;

    private IEnumerator Screenshot()
    {
        yield return new WaitForEndOfFrame();
        Texture2D texture = new Texture2D(Screen.width, Screen.height,
TextureFormat.RGB24, false);

        texture.ReadPixels(new Rect(0, 0, Screen.width, Screen.height), 0, 0);
        texture.Apply();

        string name = "Screenshot" + System.DateTime.Now.ToString("yyyy-MM-dd_HH-mm-
ss") + ".png";

        //PC
        byte[] bytes = texture.EncodeToPNG();
        File.WriteAllBytes(Application.dataPath + "/../" + name, bytes);

        Destroy(texture);

        canvas.SetActive(true);
    }

    public void TakeScreenshot()
    {
        canvas.SetActive(false);
        StartCoroutine("Screenshot");
    }
}

```

Figure 30 Script for take the screenshot