SIMULATION GAME ON THE LIFESTYLE OF FTMK STUDENT

NUR IZZATUL AMIRA BINTI MUZAFAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SIMULATION GAME ON THE LIFESTYLE OF FTMK STUDENT

NUR IZZATUL AMIRA BINTI MUZAFAR

This report is submitted in partial fulfillment of the requirements for the Bachelor of Information Technology (Game Technology) With Honors

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL: Simulation Game on The Lifestyle of FTMK Student

SESI PENGAJIAN: 2015/2016

Saya Nur Izzatul Amira binti Muzafar

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan
Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_____	SULIT	(Mengandungi maklumat yangberdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
_____	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
/	TIDAK TERHAD	

_____ _____

(NUR IZZATUL AMIRA MUZAFAR) (MS SYARIFFANOR HISHAM)

Alamat tetap:

81, Kampung Kinjang Baru

35500, Chenderiang,Perak.

Tarikh: 23/8/2016 Tarikh: 24/8/2016

CATATAN: * Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

DECLARATION

I hereby declare that this project report entitled

SIMULATION GAME ON THE LIFESTYLE OF FTMK STUDENT

Is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: _____ Date: 23.8.2016

(NUR IZZATUL AMIRA MUZAFAR)

I hereby declare that I have read this project and found this project is sufficient in term of the scope and quality for the award of Bachelor of Information Technology (Game Technology) With Honors.

SUPERVISOR: _____ Date: 24/8/2016

(SYARIFFANOR BINTI HISHAM)

DEDICATION

To my beloved mom.

The strongest person I know.

ACKNOWLEDGEMENTS

First of all, I am grateful to God for giving me strength to be able to complete this project.

I would like to thank Ms Syarrifanor Hisham for giving assistant to complete this project successfully.

I would also like to thank my beloved mother and my brothers who have been giving me support and motivation throughout my project.

I would also like to thank to my beloved friends for being very supportive throughout this project.

ABSTRACT

A simulation video game is a diverse super-category of video games, generally designed to closely simulate aspects of real or fictional reality. A simulation is a recreation of a real-world situation, which try to accurately depict real events and physics as accurate as possible. The selected approach used in development of FTMK Simulator is the Game Development Life Cycle (GDLC) where this process contain five main stages which is concepting, design, production and evaluation. The gameplay of this game is player controls the first person character and needs to level up the programming and art skill. Player also need to refill the energy to continue level up the skills and complete the task by using the energy to complete the Year and to finish the game. The FTMK Simulator is developed to provide an experience to player outside FTMK and UTeM whose objective is to closely simulate aspects of various activities of real life student in FTMK in the form of a 3D simulation game and to replicate the environment in UTeM in virtual form. This is a first-person perspective game which will gives more immersion to the player in virtual world. FTMK Simulator is aimed to give an experience to the player about being a computer science student in UTeM and to promote UTeM as well.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A simulation video game is a diverse super-category of video games, generally designed to closely simulate aspects of real or fictional reality. A simulation game attempts to copy various activities from real life in the form of a game for various purpose such as training, analysis, or prediction. Usually there are no strictly define goals in the game, with players instead allowed to freely control a character. There are several varieties of simulation games, but are probably are three that are very well known, such as war games, business games, and role play simulation.

1.1 Project Background

FTMK Simulator is a simulation role playing game which simulates real environment of FTMK (Fakulti Teknologi Maklumat dan Komunikasi) and FTMK students where the player controls the first person character and play as a freshman Computer Science student by completing the task as the students of FTMK in real life. Player needs to level up both Programming and Art skills in order to get good grades at the end of the game by attending classes and practicing the both skills outside class by using the laptop provided in the players' inventory. The target groups of this game is young adults between 18-20 years old, and students from another courses outside FTMK and UTeM (Universiti Teknikal Malaysia Melaka) in general.

1.2 Objectives

- To simulates aspects of various activities of real life student in FTMK in the form of a game
    - The activities including attending classes, navigating through the FTMK environment, using the water cooler and vending machines, completing the assessments and levelling up both primary skills needed by computer science students.
- To study the application of virtual world by simulation game
    - Promoting the uniqueness of FTMK buildings using virtual world environment.
- To develop a game that simulates the environment of FTMK in real life
    - This includes the buildings, the area of FTMK surroundings and the objects that are available inside the building including computer labs and vending machines.

1.3 Goals and Genres

FTMK Simulator aims to entertain and promoting FTMK to other students from another faculties outside FTMK and UTeM students as well as to give the players sense of curiosity about the lifestyle of FTMK students in general. The genre of the game is role playing simulation game where the player plays as computer science student and need to develop the core skills needed in the game.

1.4 Game Features

Target groups of the game is students outside FTMK, where most of the students never visited FTMK before and computer science outside UTeM, to give a clear view about the lifestyle of computer science students in UTeM. This game also aim for the 18-20 young adults especially SPM leavers and promoting UTeM to them.

The game requires player need to attend the classes in order to level up the Programming and Art skills. Both skills will determine the grade players will achieved at the end of the game. If the player missed the classes, they still can level up their skills, but the progression is slower than attending the class. All three ways of levelling up the skills will require the Energy consumptions of the players. The higher the progress bar for each skill, the higher the grade.

1.5 Conclusion

The final version of the game gives more clear view about the iconic buildings of FTMK and the lifestyles of FTMK students to the player especially from another courses and universities. The FTMK Simulator is a game that aim to provide an experience to the player outside FTMK and UTeM whose objectives is closely simulates aspects of various activities or real life student in FTMK in the form of 3D role playing simulation game and to replicate the environment in FTMK in virtual form. This first person perspective game which gives more immersion to the player in virtual world. Literature review about the comparison of FTMK Simulator and similar games will be discussed in the next chapter.

CHAPTER 2

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

A simulation is a recreation of a real-world situation, which try to accurately depict real events and physics as accurate as possible. Role-playing game is a game in which players assume the roles of characters in a fictional setting. The genre has an almost two decade history and is represented by hundreds of commercial titles. It is most commonly found in First-Person Shooters and Racing Games, and to lesser extent in other genres, such as RPGs and 3D Platformers. In order to provide a wider context for such work this paper surveys 4 separate simulation titles. The titles are compared the mechanics and game setting.

2.2 Genre

FTMK Simulator is a simulation role playing game where this game simulates real environment of FTMK and FTMK students where the player controls the first person character and play as a freshman Computer Science student by completing the task as the students of FTMK in real life.

2.3 Existing Games

There are four games that have been developed which is Home Improvisation, Sortie En Mer, Goat Simulator and Kerbal Space Program. All these games are simulation game in general.

Home Improvisation is a furniture-building simulator that lets player decorate an entire house. This game is developed for Microsoft Windows and OS X. Using the entire furniture's of IKEA products as a base, it is basically promoting the IKEA products to the player. The game is developed for PC platform as well. Kerbal Space Program is a space flight simulator developed by Squad in Unity for Microsoft Windows, OS X, Linux, PlayStation 4, Xbox One, and Wii. It is a game where the players create and manage their own space program to conquer space. Sortie En Mer is a short and intense full motion video game in which player are knocked off their yacht and must fight to stay alive for as long as possible. Goat Simulator is a third-person perspective action video game developed and published by Coffee Stain Studios. It was released for Microsoft Windows via Steam, OSX, iOS and Android.

2.3.1 Comparison of Existing Games

Games/ Games	Home Improvisation	Sortie En Mer	Goat Simulator	Kerbal Space Program
Mechanics	- Furnitures from IKEA products -Top-down view -Single player and multiplayer	- First person -Single player game - Full motion video game -sea environment	-Third person perspective -single player -multiple game modes -glitches are left inside the game for entertainment	-single player -multiple game modes -space exploration -historical events can be recreated and their accomplishments mimicked(e.g Apollo program)
Genre	-furniture-bulding simulaton	-FMV -simulation	-action -simulation -sandbox	-space flight simulation -sandbox
Gameplay	-building furnitures and decorate house -none of the furnitures comes with instructions -toolbox are provided -player freely to assemble	-scrolling the mouse in order to stay on surface of the sea -temperature of the sea water is displayed -the time player survived will	-player acquire points by performing act of chaos -player can jump, drag an bash and object and run -freely to explore the game world	-construct rockets, aircraft, and spaceplanes to complete player-set or game directed mission -avoid various opportunities for failures(e.g lack of fuel, stuctural failure)

	the furniture with their own creativity	be displayed at the end of the game		
Platform	-Windows(via Steam) -OS X	-Android -HTML -Chrome -Google Analytics	Windows(via Steam) -OS X, -Linux -iOS, -Android, -Xbox One, -PS3 and PS4	-Windows(via Steam) -OS X

2.4 Project Methodology

The selected approach used in development of FTMK Simulator is the Game Development Life Cycle (GDLC) where this process contain five main stages which is concepting, design, production and evaluation.

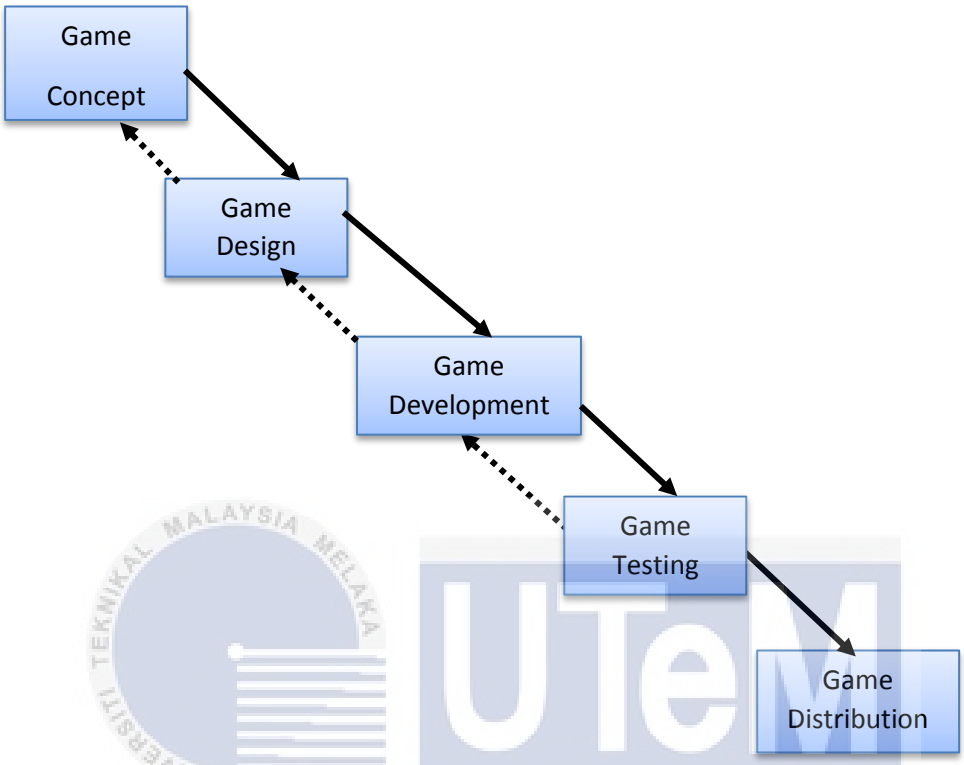

Figure 2.1: Game Development Life Cycle (GDLC)

Based on Figure 2.1, during concept phase, the goals that are covered in this game is set, established the instructive mechanism, across which knowledge will be transferred to the player. This phase also the competences and knowledge areas are covered and determined, and also to create storyboard and concept art for the game. For design phase, all digital resources needed by the game engine for the creation of FTMK Simulator are searched and created. These digital resources includes: 2D illustrations, 3D models, map, objects, materials, surfaces, sounds and music. During production phase, game development are started including the layout, events and shader, designing the game play and integrated all above elements with menus, options etc. Last phase is for evaluation or test play where the game is tested in aspects of technical, knowledge, absorption, usability and usefulness to obtain efficiency statistic and to maintain the game.

2.5 Conclusion

In literature review, there are four different games with same genre which is simulation game are compared according to their mechanics, genre, gameplay and the game platform. This chapter also describe the project methodology of the project development. The four games in literature review will be analysed and compared with FTMK Simulator in the next chapter, as well as the technical, software and hardware requirement, project milestone and schedule.

CHAPTER 3

ANALYSIS

3.1 Requirement

In game development and game optimization, a requirement is a documented physical and functional need that a game must be able to perform. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a game for it to have value and utility to the player. A requirement specification refers to an explicit set of requirements to be satisfied by a product.

3.1.1 Project Requirement

In this chapter, four games that are stated in literature review will be discussed and analysed based on the player roles, gameplay, victory condition, core mechanic, user interface features, and camera models.

The player role in Home Improvisation is to assemble the furniture and decorate an entire house using those furniture. In Sortie En Mer, the player role is to survive above the surface of the ocean as long as possible before they get drowned. In Goat Simulator, player plays as a goat and perform chaotic action as much as possible to gain points. In Kerbal Space Program, player role is to construct the rocket and spaceplanes for the Kerbal so they will be able to launch it to the space and explore it.

Gameplay in Home Improvisation is simple, is assembling the furniture and decorate the house with the furniture. There are no instructions for the furniture so player are freely to assemble it with their own creativity. In Sortie En Mer, player need to stay on the surface of the ocean as long as possible to avoid getting drowned. Player will use mouse to scroll the avatar to stay on float. In Goat Simulator, player controls a goat, and freely to explore the game's world, a suburban setting, as a goat, and jump, run, bash things, and lick objects. The game features a scoring system whereby doing tricks or other actions earns points. Small gold goat statues are hidden in the game's world. Collecting this allows player to restart the game with various modifies in play, such as changing the goat model. In Kerbal Space Program, player controls a nascent space program operated by Kerbal and construct the rockets, aircraft, and spaceplanes out of a provided set of components and launching them form the in-game space centre's launch pad. Player also able to complete player-set or game directed mission while avoiding various opportunities for partial or catastrophic failure such as lack or fuel or structural failure. Players control a space craft and keep the rocket pointed at a player specific direction to keep a constant attitude.

In Home Improvisation, there is no victory condition to end the game because it basically an endless game where player will keep on assembling and decorating the house with unlimited amount of furniture. In Sortie En Mer, the game ends when

player finally drowned and the time they had survived are displayed. In Goat Simulator, the game has no victory nor lose conditions where it also an endless game where player will keep on performing stunts and actions.

Core mechanics of Home Improvisation is the furniture are delivered to the player in sequence after the player completed assembling the previous furniture. The furniture comes with no instructions, and the furniture are lamps, chairs, coffee tables and other IKEA products. Toolbox are provided so the player can customize the existing furniture. In Sortie En Meer, the temperature of the ocean will be displayed and after the game ends, the time for the player drowned will be shown. For Goat Simulator, the goat can be customized into various types of goat such as demon goat when player collecting the gold goat statue inside the game world. The Easter egg also included in the game. The points are given to the player when player perform several of stunts and higher points are given when the stunts are combined. The Kerbal Space Program contains parts of the rockets, aircraft and spaceplanes to be constructed. A fuel also available in the game for the rockets to be launched. The player will get an achievements after completing the missions. Home Improvisation use the top down view of the camera, while in Sortie En Meer use the first person perspective view of the camera. Goat Simulator uses the third person view for the camera.

3.1.2 Technical Requirement

The hardware used in the game is PC, and using mouse and keyboard as a main input controller.

3.1.2.1 Software Requirement

- Game Engine
    - Unity 4.6 for game's backend activity.
- Game Art
    - Adobe Audition CS6 for audio editing.
    - Autodesk Maya 2016 for creation of 3D assets.
    - Autodesk 3d Studio Max 2014 for creation of 3D assets.
    - Adobe Photoshop CS6 for 2D graphic editing.
    - Adobe Flash CS6 for 2D graphic creation.
    - Unity 4.6 for 3D art creation.

3.1.2.2 Hardware Requirement

- Development
    - PC as a platform to develop the game and will be deployed on the same platform.
    - Keyboard as an input device for editing and testing the game.
    - Mouse as a main input device for testing and editing process.
- Player Interaction
    - Keyboard to control the movement of the player in the game
    - Mouse to control the camera of the player's character
    - PC as a main platform to play the game.

3.2 Project Schedule and Milestone

The project started on 21 February and ended on 22 February and the status is complete. Next phase is an Art concept started on 22 February and ended on 24 February. The art concept is completed during the phase. Storyboarding started on 24

February and ends on 1 March and the status is complete. The development of the game starts on 1 March where all the level design phase, creation of assets, programming, behaviour and tools and ended on 10 May and is not complete because these elements need to be reconstruct after the playtesting phase. Playtesting started on 10 May and ended 3 Jun and the still in the alpha phase where the game is not yet fully tested. The deploying of the game started on 3 Jun and the project will end on 27 July for full product and documentation.

Table 3.1: Project Milestone

START DATE	MILESTONE	END DATE
21-Feb	Project Start	22-Feb
22-Feb	Art concept	24-Feb
24-Feb	Storyboarding	1-Mar
1-Mar	Writing	5-Mar
5-Mar	Level design	10-Mar
10-Mar	Modelling and animation	3-Apr
3-Apr	Programming, behaviour and tools	10-May
10-May	Playtesting	3-Jun
3-Jun	Deploying the game	27-Jun
27-Jun	Project End	

Table 3.2: Gantt chart of Project Activities

TASKS	WEEK														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Getting idea and first meeting with supervisor															
Preparing the proposal															
Submission of proposal															
Designing the gameplay															
Constructing the architecture															
Constructing the flow board and Pseudo code															
Constructing the hierarchy of challenge															
Designing the game rules and user interface															
Project development															
Play Testing															
Final presentation															
Preparation of project report															
Submission of project report															

3.3 Conclusion

There are four games that were compared during literature review and those games are compared in terms of the mechanics, gameplay, features and the technology used in the game. During analysis, FTMK Simulator is compared with those four games in similarities of the elements and the differences. The technology and software used in the development of the game also included in analysis. The project milestones and progress are explained in this chapter. Next chapter will explain the game design process.

CHAPTER 4

DESIGN

4.1 Introduction

Game design is the art of applying design to create a game to facilitate interaction between players. Game design creates goals, rules, and challenges to define a sport, table top game, casino game, video game, role-playing game, or simulation that produces desirable interactions among its participants. In this chapter will explain about the game architecture, the game design process, the challenge, and flow of the game as well as the interface of the game. These elements are included in FTMK Simulator.

4.2 Game Architecture

In game architecture, there are three basics that need to be established, which is the inputs, time passes, audio and the rendering capabilities. Elements of the game architecture including the major subsystems, main loop, time, game phase, and entities. Below is the game architecture that was applied on developing FTMK Simulator.

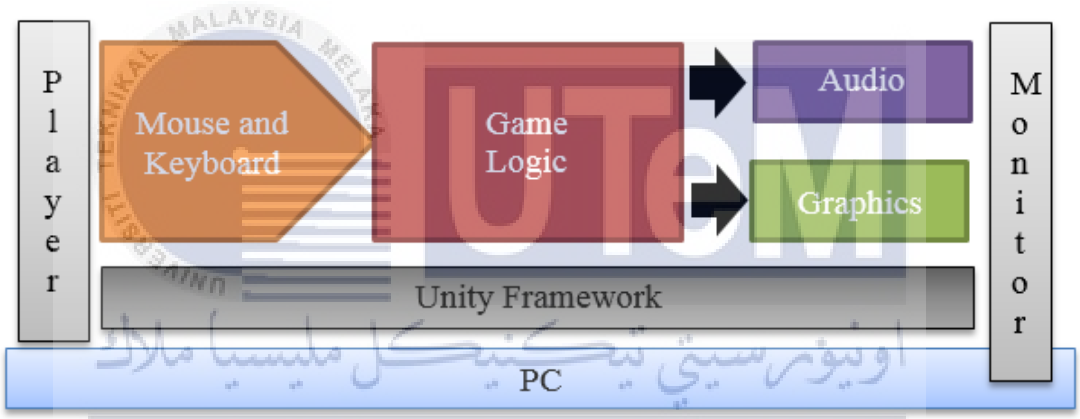

Figure 4.1: Game Architecture of FTMK Simulator

Based on Figure 4.1 above, the game architecture of FTMK Simulator runs in 4 phase which is the input, the game logic, audio and graphics and the output. The input is received from the player and all recognized input will be run in game logic where all the mechanics and technical behavioral aspects is calculated and processed. The audio and graphics are retrieved from the game logic process to be displayed to the player in output stage. The user input, game logic and audio and video integration process handled by the Unity framework. All these elements will be run in PC platform for Windows and OS X.

4.3 Game Design

Game design is the process of designing the content and rules of a video game in the pre-production stage and designing the gameplay, environment, storyline and characters in the production stage. In FTMK Simulator, there elements are crucial as a starting point for the development of the game.

4.3.1 Gameplay

Gameplay is the specific way in which players interact with a games, and in particular with video games. Gameplay is the pattern defined through the game rules, connection between player and the game challenges and overcoming them, plot and player's connection with it.

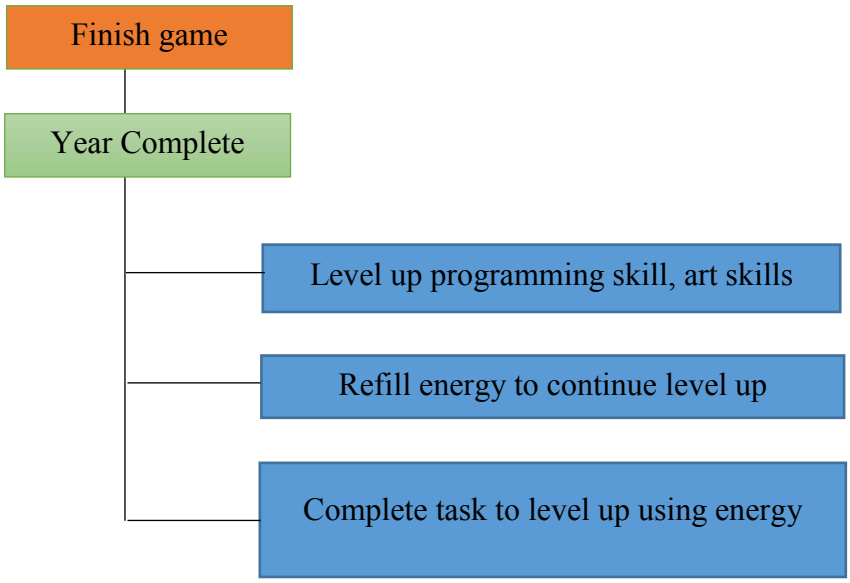

Figure 4.2: Hierarchy of Challenge

This figure shows the hierarchy of challenge of FTMK Simulator where the player needs to level up the programming and art skill. Player also need to refill the energy to continue level up the skills and complete the task by using the energy to complete the Year and to finish the game. Player controls the first person character to navigate around FTMK buildings. Player needs to get higher level of skills in order to get good grades at the end of the game. Player will get lower grades if they not get enough skills in the game.

4.3.2 Core Mechanics

In FTMK Simulator, there are two skills to be levelled up, Programming and Art skill and player require enough Energy to complete the tasks in both skills. There is an inventory where the player can use their laptop to level up the skills and the timetable to check the time for the classes. Real-time clock also available to the player. Vending machines can be used to refill the Energy bar. The game is first person view of camera and involves character development which is the skills progression (programming and art).

4.3.3 Flowboard

A flowboard is a cross between a flowchart and a storyboard. It combines two ideas to document the structure of a game. During the design phase of FTMK Simulator, the structure of the game are documented including the links among gameplay, modes and shell menus.

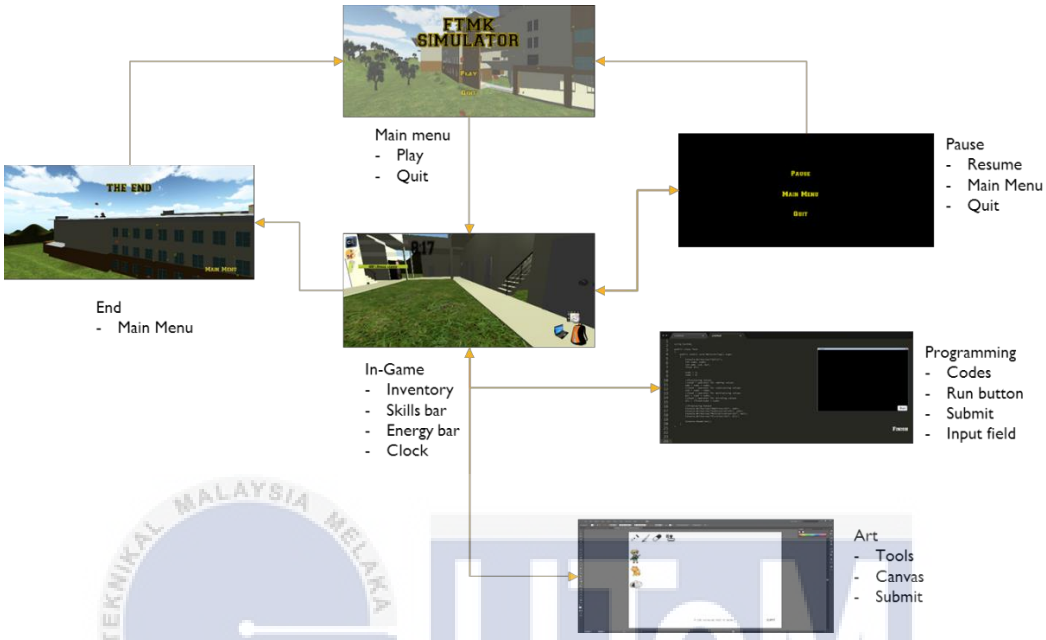

Figure 4.3: Flowboard of FTMK Simulator

Based on the flowboard above, the game starts with the main menu, where it consists of the Play and Quit button, when the player clicked play, the game starts. In-game interface consists of inventory, the skills bar, the Energy bar and the clock. When the game pauses, it stops the clock and player is given three buttons which is Resume, Main Menu and Quit. Player can resume the game when they choose Resume button, while Main Menu button restarts the game and Quit button to quit the game. In Programming interfaces, player inputs the answers and exit the interfaces by clicking Submit button. Same goes to Art interfaces. The game ends when the player managed to complete the goals provided.

4.3.4 User Interface/Interaction Design

User interface design is the design of user interfaces for machines and software, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing player goals.

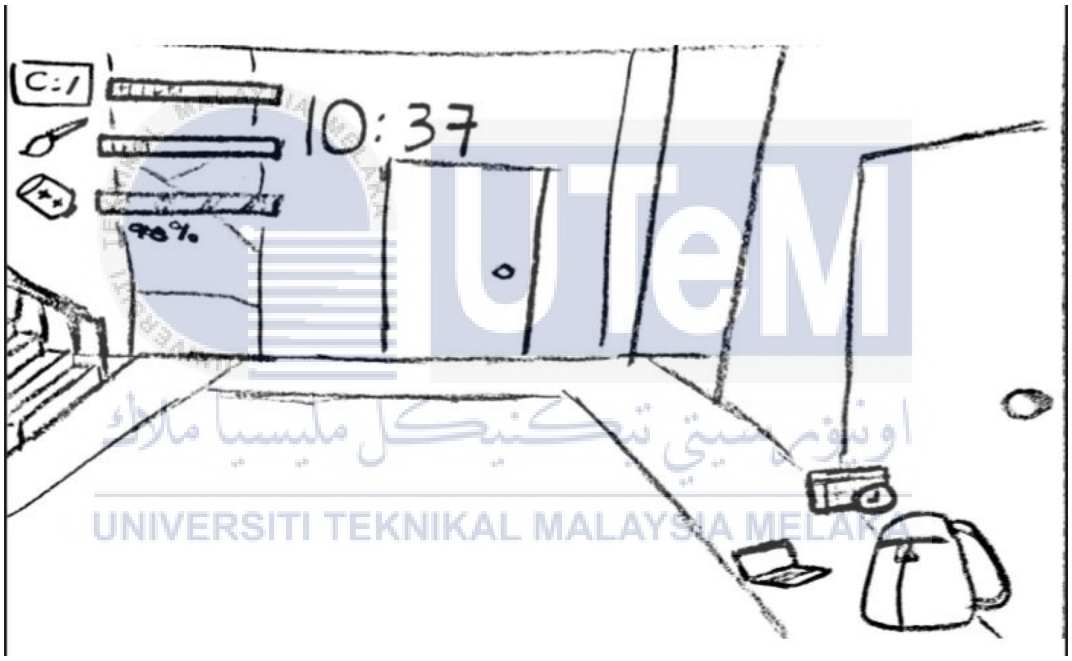

Figure 4.4: In-Game Interface Concept

Figure 4.5: In-Game Interface

During concept stage, the interface including the icons is sketched as an outline to utilize the camera view when the player plays the game as shown in Figure 4.5. In FTMK Simulator, the interface was designed in such a way to provide efficiency, and satisfaction which can be seen as a quality factors of usability. The interface including the real-time clock, the Programming and Art skill Bar and Energy Bar on top left of the game, backpack as an Inventory consist of timetable and laptop for player to interact. When the design of the interface is finalized, it sketch of the interface was transformed into 3D and used inside the game.

4.4 Game Art

Game art design is the process of creating the artistic aspects for video games. Video game art design begins in the pre-production phase of creating a video game. There are several elements of game art that are used in FTMK Simulator including the game world, the camera model and also the audio and sound effects

4.4.1 Game World

The game world are almost always implemented as some sort of simulated physical space. The environmental dimension describes the world's appearance and its atmosphere, which are used when creating the game world for FTMK Simulator.

Figure 4.6: FTMK building

FTMK building is chosen as the game world for this game, it consists of several buildings and various design in some of the buildings, to reach the objectives of promoting the uniqueness of FTMK building. The creation of the FTMK building is started during the production stages, where the pictures is used as a framework for this model.

4.4.2 Camera Model

Camera model in this game aims at controlling a camera to display a view of a 3D world where the purpose is to show the action at the best possible angle for the player in the game. This game uses the first-person perspective where it is rendered from the viewpoint of the player's eye. First person view is chosen to give more immersion to the player as well as it does not require sophisticated animations for the player's avatar, nor do to need to implement a manual or automated camera-control scheme as in third-person perspective. The player uses mouse to control the camera which is for easier aiming in first-person perspective.

4.4.3 Audio/Sound Effects

The audio used in this game consist of background music and sound effects. The background music chosen for the game is instrumental because it suitable with the college life-themed background of the game. It also gives player the curiosity about the game as well. The sound effects including for footsteps, yawning, the open and closed door, the vending machines and water cooler, the energy bar depletion, and the level up skills. All of these sound effects are used to give more immersion to the player in the game.

4.5 Conclusion

Game architecture contains five main parts in order for the game to be able to run. The core mechanics, the challenges are determined in this phase. The flowboard shows the flow of the game and navigation throughout the game world. The interface designed in the specific ways for the player affordance. FTMK building are modelled

and used in as the game world along with the terrains and other accessories. The first person camera is used for FTMK Simulator. Background and sound effects are used in the game. In the next chapter will discuss about the implementation of the game.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

Implementation can be seen as a sandbox where things will be tried out and pruned to meet the required objectives. The resources for this game that already available are compared with the resources that the project require. This resources including the assets, the hardware and software, as well as the software and hardware limitations to handle the resources. This chapter will shows how the game is programmed, the music is composed and the creation of game arts before all the resources are collaborate for testing and debugging.

5.2 Creation of Game Art

The basics of a 3D production pipeline consists of several steps, firstly is during pre-production. During this phase, the FTMK environment is chosen as a direction of the game first taken form. This step is important to help developer determine the concept for style, look, sets, characters and much more. Next step is the modelling, where the assets to be used in this game are all modelled from scratch, or adapted from other models, using variety of techniques to meet project requirements. This phase takes all of the concepts from pre-production and brought them to life. Assets are modelled with a style set forth in the pre-production phase. During painting and texturing, the models and assets is textured to give colours and materials to the models. Next step is lighting, where the light elements of the scenes are controlled. Lighting gives realism to the models and assets. The last step is the rendering, where all the assets and models are imported to Unity and rendered.

5.2.1 Production of Graphics

The game is developed into a 3D environment which the scene and model is created in Maya. For the game world, FTMK building is outlined using the top view using Google Earth as a blueprint. The blueprint is important as a wireframe to gives an idea to create a foundation of the model for next steps, which is the modelling process.

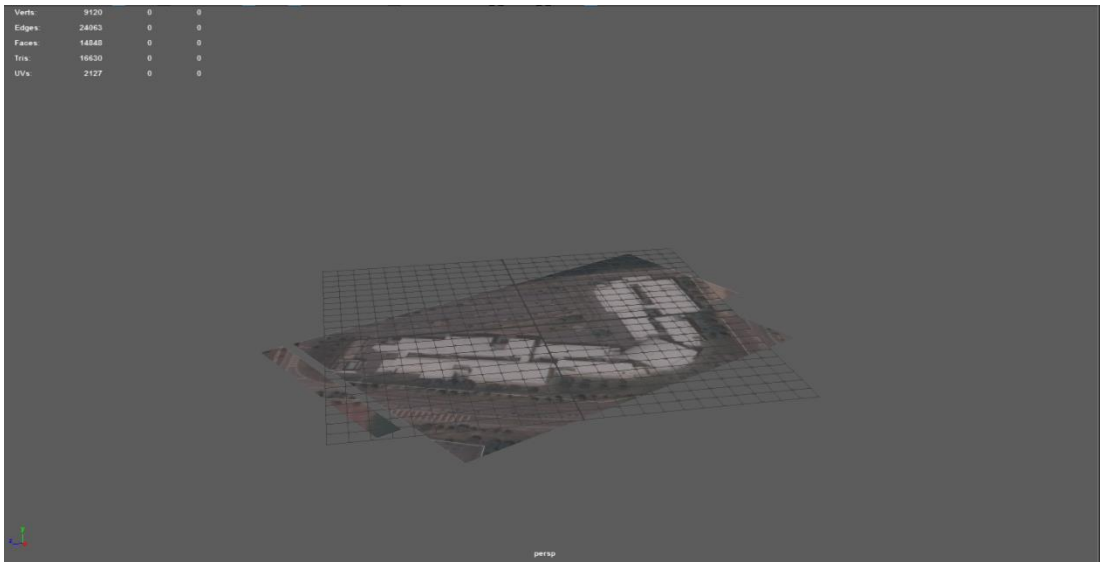

Figure 5.1: Top view of FTMK building

The next step is the modelling, where FTMK is modelled from scratch. The modelling starts with the basic grey polygon, and is modelled part by part.

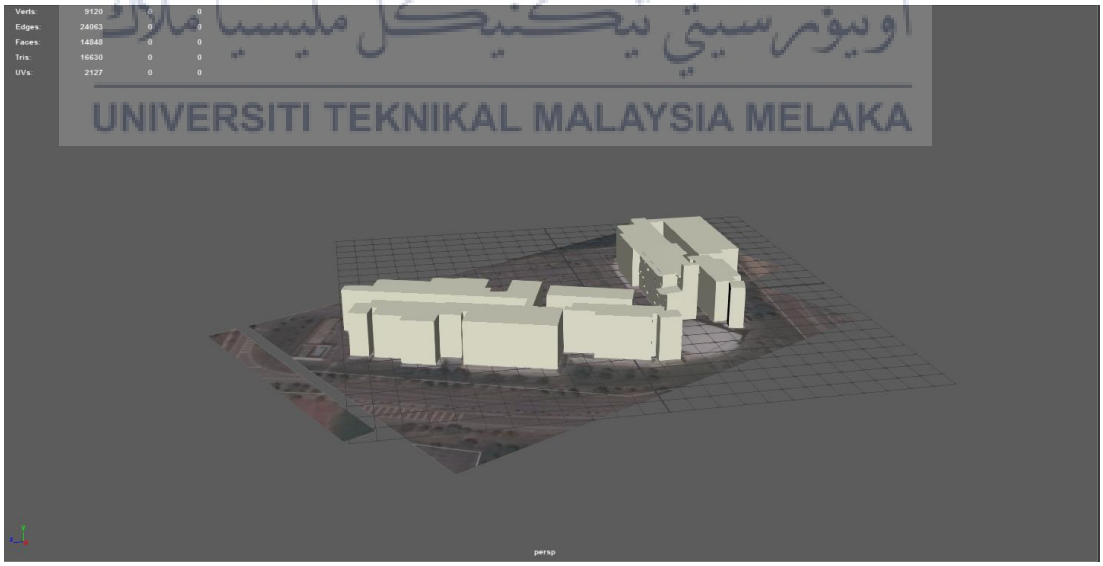

Figure 5.2: Basic model of FTMK building

Next steps is where the model surfaces are divided into parts, where each of the surface in each model is given the texture and painted as similar as possible to FTMK building.

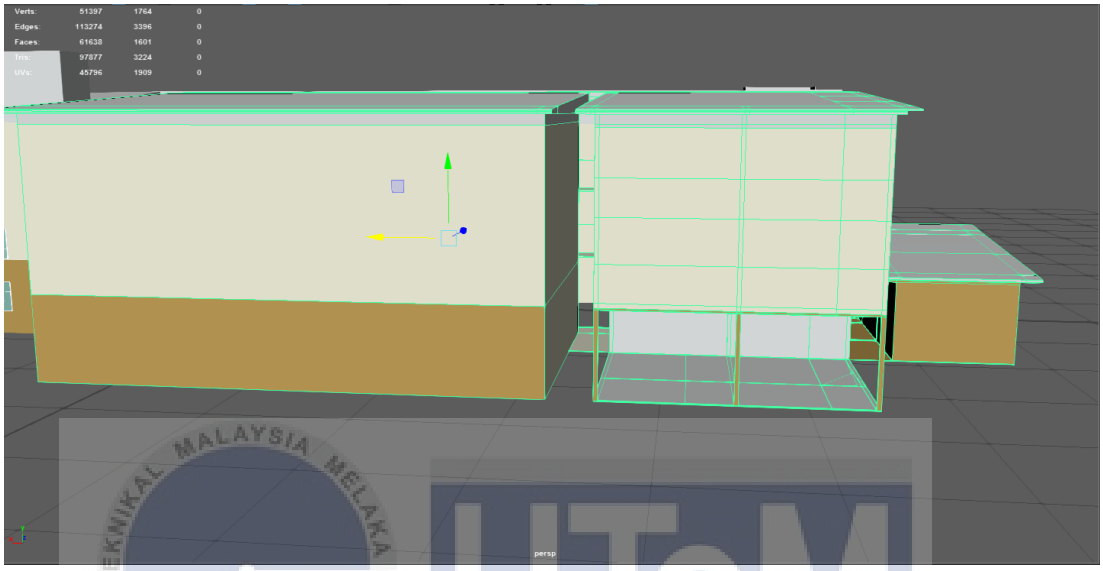

Figure 5.3: Subdivisions of model of FTMK building

The windows, stairs, railings and doors are added to the model to give more realistic view and the doors will be used later in the game.

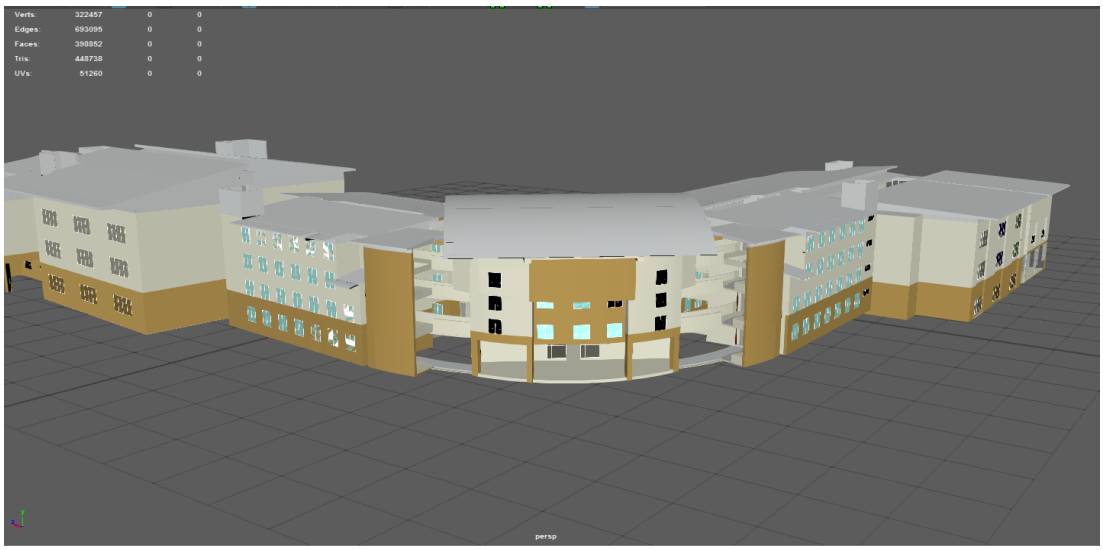

Figure 5.4: Complete model of FTMK building

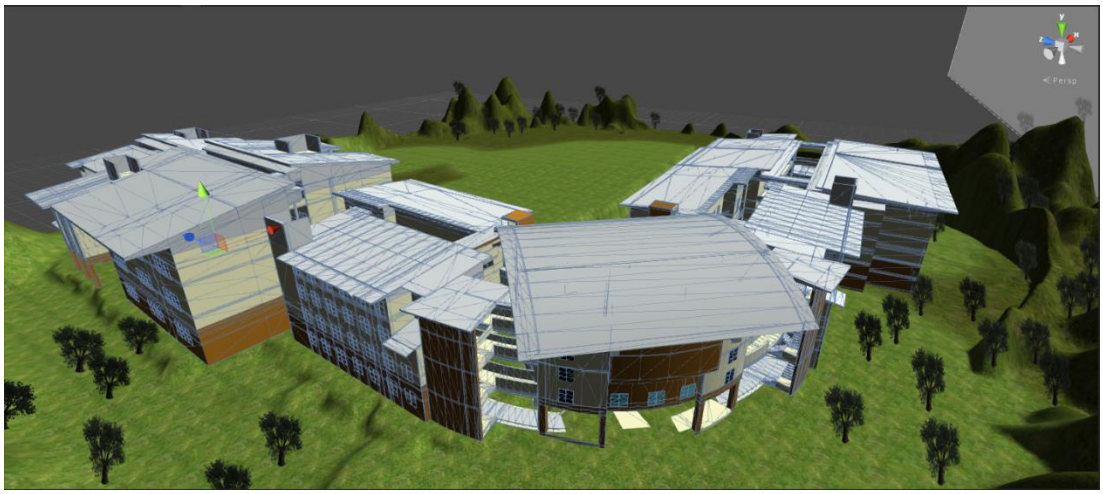

Figure 5.5: Model of FTMK building exported in Unity

The model were exported to Unity and arranged in the game world. The terrain is created using terrain tools provided in Unity and decorated with 2D grass texture and foliage.

5.2.2 Production of Audio

Digital audio is based on the conversion of sound waves into discrete values, obtained by sampling the wave multiple times a second, and storing the amplitude of the wave at each sampling moment. The musical features of video game mostly depend on the genre the video game belongs to. FTMK Simulator utilize this feature by using both music and sound effects to enhance the mode of the player. Instrumental music is used as a background music for the main menu scene. The music is downloaded from free source and is edited in Audition to remove any echoes and background noises. Many action and simulation games include sound effects recorded from real world situations in order to simulate environment. FTMK Simulator uses this advantages to improve the scenery of the game, as well as an immersion to the player.

The sound effects of the game including the footsteps and cracking door are also downloaded from the free sources and edited using equalization, mixing, reproduction and reinforcement technique.

5.3 Integration of Game Components

The game is developed into a 3D environment and will be developed in Unity. Unity will be responsible for both the construction of the game within its framework. Construction of the game including the physics, collision detection and handling, input recognition, object creation and transform manipulation which is the position and rotation of game objects, scene integration which is the transition of one scene to the next, and model attachment which represents game objects with 3D models from external programs.

The transform manipulation is implemented in this game to open and closes the door. The code snippet is shown as below:

```
public void ChangeDoorState()
{
        open=!open;
}
void Update ()
{
        if (open)
        {
                Quaternion targetRotation = Quaternion.Euler (0, doorOpenAngle, z);
                transform.localRotation = Quaternion.Slerp (transform.localRotation,
        targetRotation, smooth * Time.deltaTime);
        }
        else
        {
                Quaternion targetRotation2 = Quaternion.Euler (0, doorCloseAngle, z);
                transform.localRotation = Quaternion.Slerp (transform.localRotation,
        targetRotation2, smooth * Time.deltaTime);
        }
}
```


Figure 5.6: Snippet Code of Door Rotation on Y-Axis

To make the above code to be able to run, the player needs to have a collision detection with the door so the door will successfully open. The ChangeDoorState() will check whether the collision between the door and the player. If the 'open' variable is activated, the lines in Update() will enables the door to rotate on the y-axis thus open and close the door. The next code snippet enables the player to be able to interact with the door:

```
if (Input.GetKeyDown (KeyCode.Mouse0))
{
        Ray ray = new Ray(transform.position,transform.forward);
        RaycastHit hit;
        if(Physics.Raycast(ray,out hit,interactDistance))
        {
                if(hit.collider.CompareTag("Door"))
                {
                hit.collider.transform.GetComponent<doorscript>().ChangeDoorState();
                }
        }
}
```

Figure 5.7: Snippet Code of Collision Detection

Based on code snippet above, the player can only interact will the door if the player is close to the door, by means within the range that is set in Inspector on Unity. The code will recognize the selected door from the tag "Door" that has been assigned to every door in the game. The code will run the component from another script 'doorscript' from Figure 5.6.

5.4 Game Configuration Management

Unity3D's build setting simplify the process of transferring the game into a PC platform. After completing the project, or during any intermediary step for testing, the game will be deployed on PC.

5.4.1 Version Control Procedure

One of the central activities in this project is the constant addition, deletion, and modification or source code. In addition to coordinating access, whenever modifications is made, the version control will maintains a separate version of each file. The first step in version control for this game is designing the directory layout. The strategy for implementing the design was map out. Next steps is finding a home for shared code. The directory layout of modules in the project is visualized. The next steps is monitoring and refining usage patterns. The usage patterns constantly monitored. Version control also used to address any bug in the game. Lastly, during the integrate and test phase, where the game will tested depending on the amount of code and the overall length of the project. This game currently in alpha phase where the game will be demonstrate to prove that this game is technically possible.

5.5 Implementation Status

During the development phase on Week 4, there a several major activities that involved including level design phase, creation of assets, programming, behaviour and tools. The 3D assets are created and exported to Unity on Week 5. The level design phase are conducted for every complete element to test the functionality and the synchronization of the gameplay in Week 5. The programming and behaviour are created in Week 6, where all the mechanics and gameplay are implemented during that duration of time and completed on Week 11. After all the elements are complete, the elements are implemented and tested on Week 11. However some elements such as 3D assets caused the prototype to be delayed on completion because of the technical issues occurred during the creation of 3D assets.

5.6 Conclusion

This chapter showed how the game is programmed, the music is composed and the creation of game arts before all the resources are collaborate for testing and debugging. The basics of a 3D production pipeline consists of several steps, which is pre-production, modelling, painting and texturing, lighting, and rendering. The game is developed into a 3D environment which the scene and model is created in Maya. FTMK Simulator uses both background music and sound effects during this phase. Unity are responsible for both the construction of the game within its framework. In the next chapter, this report will be discussed on the testing and evaluation phase for FTMK Simulator development.

CHAPTER 6

TESTING AND EVALUATION

6.1 Introduction

Testing and evaluation is the set of practices and processes used to determine if the product under examination meets the design, if the design correctly reflects the functional requirements, and if the product performance satisfies the usability needs of personnel in the field. The trend and competency of testing is changing. Testing now is not only limited just to find bugs but has a wider scope and is required right from the beginning of the project when the requirements are not even finalized. In this chapter, FTMK Simulator was tested and evaluated according to this five stages which is unit testing; to ensure that each individual components that are tested independently operates correctly without other game components. Next is integrated testing which combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules or functions. Next is system

testing, the first level in which the complete game is tested as a whole. The goal at this level is to evaluate whether the game has complied with all of the requirements and to see that it meets quality standards. Lastly is the acceptance testing, is conducted to determine to ensure whether the game is ready for release. The testing strategy is the general approach to the testing process rather than a method of devising particular system or component tests. The testing strategy that has been used in the development of FTMK Simulator is Top-Down Testing. This strategy is used to detect an unnoticed design errors to avoid extensive re-design and re-implementation.

6.2 Test Plan

A test plan is a document describing the testing scope and activities. It is the basis for formality testing any software or product in a project. The purpose of this game test is to evaluate the state of the game and the quality of the gaming experience in terms of the technical, content and functionality of the game. This test plan is in production stage where the game is developed and tested in aspects of technical, knowledge, absorption, usability and usefulness to obtain efficiency statistic and to maintain the game. The target groups of this test plan is the game developer and several students aged 19-22. The testing methods that are used is bug hunting. Bug hunting is used to locate flaws in FTMK Simulator and reporting them. The game feature that were tested are game mechanics and the interfaces functionality.

6.3 Test Implementation

This testing was conducted in closed playtest, where the game developer and several players are chosen to tests a game for bugs and design flaws. The bug hunting process including finding the new bug and recreate them. Next step is to report the actual result of the bug appeared, the steps to replicate the bug if the bugs are too difficult to be fixed, the severity of the bug which might affect the gameplay and user experience of the player and lastly, summarizing the bug report. The bug was documented using the bug log and survey that were updated by the game developer to track down the bug detected in the game, as well as to update status of the bug fixes. This chapter includes some test case for the game to check if the game works properly in various situations. This report will gives two examples of for two different situations.

6.3.1 Test Case 1

The objective of the Test Case 1 is to check the clock functionality in the game.

Test Case	: This test will check if the clock is working correctly.
Test Procedure	: Add a text box on canvas. Add scripts in text box. Run scene.
Expected Result	: Clock are working correctly.
Actual Result	: The clock starts with 00:00 instead of 07:00.
Comment	: Need to add checking in the scripts for the objects that have particular script. Need to check whether the arithmetic is correctly coded.
Conditional Test	: Again run scene.

Expected Result : The clock is starting at 07:00.

Actual Result : The clock works perfectly.

Accuracy : Accuracy depends on the arithmetic algorithm in the script.

6.3.1 Test Case 2

The objective of Test Case 2 is to figure out the arrangement of the canvases in the game.

Test Case : This test will check if the canvases displayed and working properly.

Test Procedure : Add a multiple canvases with different user interface for each interaction.

Expected Result : Canvases are displayed properly.

Actual Result : Some of the canvases are displayed at the same time.

Comment : Need to add checking in the scripts for the objects that have particular script. Need to check whether the correct canvases are correctly coded into the script for each interaction.

Conditional Test : Again run scene.

Expected Result : The canvases are displayed correctly.

Actual Result : The canvases displayed according to the interaction executed by the game.

Accuracy : Perfectly accurate.

6.4 Test Results and Analysis

The bug log contain bug identification number (ID), which is unique for each bug that has been reported, the date when the bug report is submitted, and the summary or short description of the bug.

Next is the severity of the bug which indicates how damaging the bug is. There are three indicators of bug severity used in this bug log, which is major, minor and trivial. Major indicates major loss of function of the game, minor is the minor loss of function, or other problem where easy workaround is present while trivial is a cosmetic problem for example misspelled words or misaligned text. Priority decides in which order the bugs should be fixed in. The highest priority is P1 and the lowest is P3. Priority is determined by combining severity with the frequency of the problem. The resolution indicates what happened to the bug and is set by the developer. There are five resolution that has been used in the bug log which is fixed, invalid bug, the bug that will never be fixed (wontfix), the bug that cannot be fixed (cantfix), and the bug where all attempts at reproducing the bug were futile, and reading the code produces no clues as to why the described behaviour would occur (worksforme).

Bug#	Submit date	Summary	Severity	Priority	Resolution	Status
1	22/03/2016	the clock does not starts from the variable set from the inspector	major	P2	fixed	resolved
2	01/04/2016	timetable from the inventory does not enlarge when the button is pressed	minor	P3	fixed	resolved
3	11/04/2016	the sun does not rotates correctly	major	P3	fixed	resolved
4	11/04/2016	player can walk through the locked door	major	P1	fixed	resolved
5	20/04/2016	the door won't closed automatically	minor	P3	worksforme	closed
6	22/04/2016	three canvas appear in same camera simultaneously	major	P1	fixed	resolved
7	02/05/2016	the buttons from art canvas does not work	major	P1	wontfix	closed
8	12/05/2016	the task canvas does not appear when the player enters the classroom	major	P1	worksforme	reopened
9	15/05/2016	health bar does not working properly after using the vending machine	minor	P3	wontfix	new

Table 6.1: A Bug Log

Table shows the bug log reported during the testing phase. There are nine bug currently detected by the game developer in FTMK Simulator. There are six bugs with major severity that are needed attention to be fixed as soon as possible and three minor bugs. There are currently no trivial bugs that has been detected. Among six major bugs that has been reported, only one bug that are currently on 'wontfix' state, which is the bug number 7, that has been documented on 2nd May 2016 and has been closed. There is also a minor bug that are on 'wontfix' which is the bug number 9, which has been reported recently. There are two bugs that are on 'worksforme' which are bug number 5 and 8, where the bug number 5 is closed for testing and bug number 8 is reopened for bug fixing.

During the survey, 20 respondent are chosen to answer the survey regarding the technical aspect of the game. 16 of the respondents are students while the remaining four respondents are from the game developers. Below are some of the results from the survey that has been conducted.

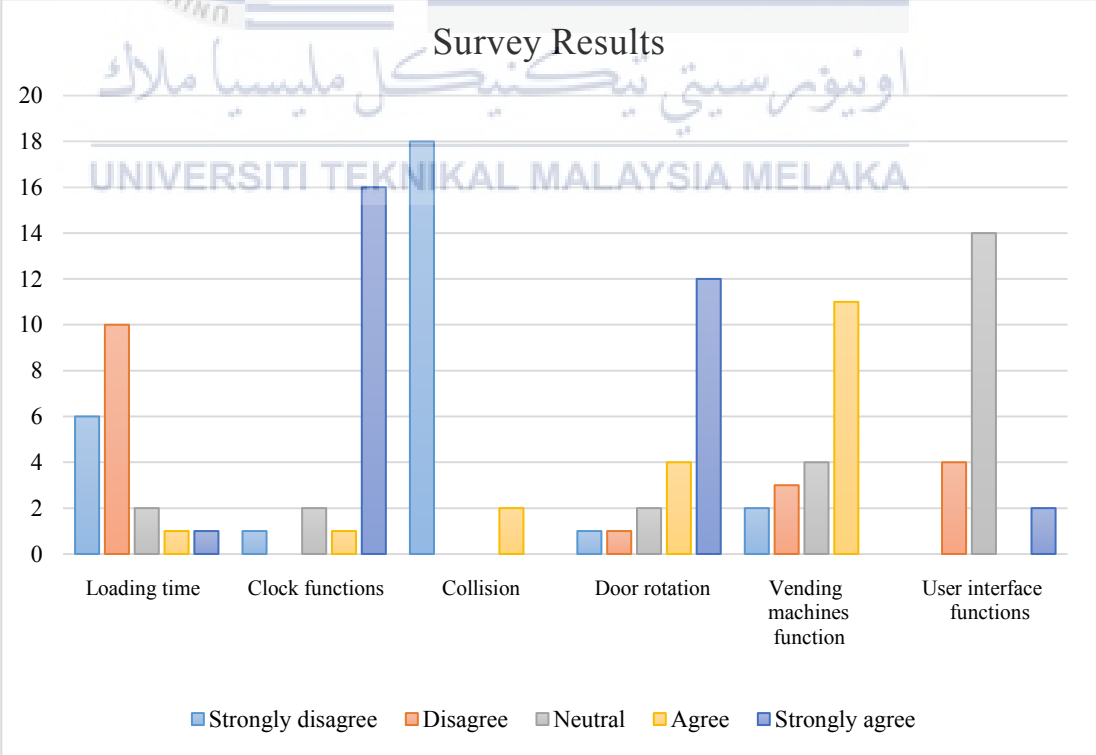

Figure 6.1: Survey results of FTMK Simulator

Figure above shows the results from the survey that has been conducted to test the technical aspect of the game. The result of loading time of the game is obtained where 10 people disagree on the game is taking too much time to load. Six respondent strongly disagree on the increased loading time of the game. The remaining two respondent agree and strongly agree about the increased loading time of the game. The result shows that the amount of time for the game to load depends on the specification of the hardware used to play the game, which is the PC.

The result for clock functionality shows that 16 respondent strongly agree that that the clock is working correctly, while two respondent remain natural about the clock functionality. The remaining results shows that one respondent agree that the clock is working, while the other one encountered a bug regarding the clock functionality. This bug is considered a major bug and will be fixed immediately.

The next result obtained from the collision detection between the player's avatar and the door shows that only two respondent encountered this bug, while the remaining 18 respondent strongly disagree that they can move through the closed door. This is considered a minor bug and will be fixed later.

Next result shows that most of the respondents do not have a problem interacting with doors. 12 respondent strongly agree that they can interact with the doors, while four respondents agree that the door can be open and closed. Two respondents are feeling natural about this, and the remaining two respondents disagree and strongly disagree on how the doors behave when they interacted with them. The bug is a minor bug and will be put for another testing.

The next result shows that 11 respondents having problems refilling their Energy when using the vending machines. Four respondents are being natural on this case. Three respondents are disagree about having problem refilling their Energy. The remaining two respondents do not having a problem at when using the vending machines to refill their Energy. This bug is a major bug and will be fixed immediately.

Lastly on the result of canvases functionality, majority of the respondent are feeling natural about this. Four people encountered a bug when interacting with the

canvases, while only two people do not having any problem interacting with the interfaces when the canvases are displayed.

6.5 Conclusion

FTMK Simulator was tested and evaluated according to this three stages which is unit testing, integrated testing and system testing. The testing strategy that has been used in the development of FTMK Simulator is Top-Down Testing. This strategy is used to detect an unnoticed design errors. The purpose of this game test is to evaluate the state of the game and the quality of the gaming experience in terms of the technical, content and functionality of the game. The testing methods that are used is bug hunting to locate flaws in FTMK Simulator and reporting them. The bug was documented using the bug log that were updated by the game developer to track down the bug detected in the game. The next activities is to conduct another playtesting (if possible) to improve the game features in FTMK Simulator.

CHAPTER 7

PROJECT CONCLUSION

7.1 Observation of Strength and Weaknesses

FTMK Simulator is aimed to give an experience to the player about being a computer science student in UTeM and to promote UTeM as well. The game will be developed in a course of eight weeks began on 10th March, 2016 and ended on May 30th, 2016 with 4 stages of development using Unity3D and Autodesk Maya as main game development tool. The status of the game is in post-production phase, where the game is still in alpha testing. The game features are limited from UTeM environment to only focused on FTMK surroundings because of the hardware capabilities and time to complete the whole project. There are also lots of major bugs found in this game. The game is might give some serious motion sickness to some players.

7.2 Proposition for Improvement

This game needs to be improved in terms of the technical, where there might be some problem regarding the specification that the game needs to be able to run smoothly in different hardware specifications. The game also needs an improvement in terms of the gameplay, where more levels needs to be included to strengthen the gameplay. The graphic refinement also needs to be reconsider because of the graphics capabilities on the workstation for better efficiency of the game. The camera controls needs to be improved to reduce the risk of nausea and motion sickness to the player.

7.3 Contribution

FTMK Simulator is developed to provide an experience to player outside FTMK and UTeM whose objective is to closely simulate aspects of various activities of real life student in FTMK in the form of a 3D simulation game and to replicate the environment in UTeM in virtual form. This is a first-person perspective game which will gives more immersion to the player in virtual world. This game gives more clear view to the player and the game developer of this game about the uniqueness, the complexity and the architecture design of FTMK buildings. With this game, more people will discover the FTMK and the lifestyle of their students about how students in FTMK.

7.4 Conclusion

The final version of the game gives more clear view about the iconic buildings of FTMK and the lifestyles of FTMK students to the player especially from another courses and universities. The game is basically meets the objectives where its goals is to simulate aspects of various activities of real life student in FTMK in the form of a game. The activities including attending classes, navigating through the FTMK environment, using the water cooler and vending machines, completing the assessments and levelling up both primary skills needed by computer science students are included in the final product. The project is also aimed to study the application of virtual world through simulation games. This objectives are completed through the modelling of the FTMK buildings and the atmosphere of the scene.

REFERENCES

Hofstede, G., de Caluwe, L., & Peters, V. (2010). Why Simulation Games Work-In Search of the Active Substance: A Synthesis. Simulation & Gaming, 41(6), 824-843. http://dx.doi.org/10.1177/1046878110375596

How the alpha phase works - Digital Service Manual - GOV.UK. (2016). Gov.uk. Retrieved 11 August 2016, from https://www.gov.uk/service-manual/agile-delivery/how-the-alpha-phase-works

Ellison, Kyle. "At Last, The IKEA Furniture Assembly Simulator Is Here". Go Explore. N.p., 2015. Web. 13 Jan. 2016.

Moss, Richard. "From Simcity To Real Girlfriend: 20 Years Of Sim Games". Ars Technica. N.p., 2011. Web. 13 Jan. 2016.

Murphy, David. "Video Game Invites You To...Build IKEA Furniture". PCMag Asia. N.p., 2015. Web. 13 Jan. 2016.

Yeates, S. (2005). What is version control? Why is it important for due diligence?. Oss-watch.ac.uk. Retrieved 11 August 2016, from http://oss-watch.ac.uk/resources/versioncontrol

Adams, E. (2016). The Dimensions of a Game World | Fundamentals of Game Design: Game Worlds | Peachpit. Peachpit.com. Retrieved 11 August 2016, from http://www.peachpit.com/articles/article.aspx?p=1398008&seqNum=3

Wallner, Phillip. "The Purpose Of Simulation". #HIST3812. N.p., 2013. Web. 13 Jan. 2016.

Learning with Computer Games and Simulations. (2016). Cited.org. Retrieved 21 August 2016, from http://www.cited.org/index.aspx?page_id=143

Lateef, F. (2010). Simulation-based learning: Just like the real thing. Journal Of Emergencies, Trauma And Shock, 3(4), 348. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2966567/

Kelly, S. & Nardi, B. (2014). Playing with sustainability: Using video games to simulate futures of scarcity. First Monday, 19(5). Retrieved from http://firstmonday.org/ojs/index.php/fm/article/view/5259/3877

Starr, P. (1994). Seductions of Sim: Policy as a Simulation Game. The American Prospect. Retrieved 21 August 2016, from http://prospect.org/article/seductions-sim-policy-simulation-game

Griebel, T. (2006). Game Studies - Self-Portrayal in a Simulated Life: Projecting Personality and Values in The Sims 2. Gamestudies.org. Retrieved 21 August 2016, from http://gamestudies.org/0601/articles/griebel

APPENDIX I

FTMK Simulator Survey

Answering this will help the developer to improve the technical aspect of the game. Please rate the play ability of the game from a scale of 1 (strongly disagree) to 5 (strongly agree)

The game is taking too much time to load

1	2	3	4	5
○	○	○	○	○

The clock starts at 07:00 when the Play button is clicked

1	2	3	4	5
○	○	○	○	○

Player can walk through the closed door

1	2	3	4	5
○	○	○	○	○

The door can be opened and closed correctly

1	2	3	4	5
○	○	○	○	○

The Health Bar does not increase even the vending machines are used

1	2	3	4	5
○	○	○	○	○

The Art and Programming interfaces are displayed correctly when the player enters the class

1	2	3	4	5
○	○	○	○	○

Do you have any comment or feedback for improvement regarding the game? Please write down below if any.

Your answer

SUBMIT

Never submit passwords through Google Forms.

APPENDIX II

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

public class InteractScript : MonoBehaviour {
	public float interactDistance = 5f;
	void Update ()
	{
		if (Input.GetKeyDown (KeyCode.Mouse0))
		{
			Ray ray = new Ray(transform.position,transform.forward);
			RaycastHit hit;
			if(Physics.Raycast(ray,out hit,interactDistance))
			{
				if(hit.collider.CompareTag("Door"))
				{
hit.collider.transform.GetComponent<doorscript>().ChangeDoorState();
				}
				if(hit.collider.CompareTag("doornull"))
				{

hit.collider.transform.GetComponent<doorscriptnull>().ChangeDoorState();
				}
			}
		}
	}
}
```

APPENDIX III

```
using UnityEngine;
using System.Collections;

public class doorscript : MonoBehaviour {

	public bool open = false;
	public float doorOpenAngle = 0f;
	public float doorCloseAngle = 0f;
	public float smooth = 2f;
	public float z=0f;

	public void ChangeDoorState()
	{
			open=!open;
	}
	// Update is called once per frame
	void Update ()
	{
			if (open) {
			Quaternion targetRotation = Quaternion.Euler (0, doorOpenAngle,
z);
			transform.localRotation = Quaternion.Slerp
(transform.localRotation, targetRotation, smooth * Time.deltaTime);
		} else {
			Quaternion targetRotation2 = Quaternion.Euler (0, doorCloseAngle,
z);
			transform.localRotation = Quaternion.Slerp
(transform.localRotation, targetRotation2, smooth * Time.deltaTime);
			//dooropen.Play();
		}
	}
}
```