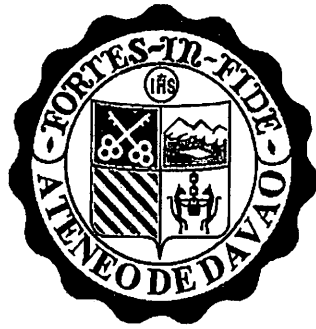


**EMILIO AGUINALDO: A SINGLE PLAYER
3D ROLE PLAYING GAME USING OGRE**



By

MICHAEL ANGELO R. CALAQUE

RONN ERIC J. LAMOSTRE

SCHOOL OF ARTS AND SCIENCES

ATENEO DE DAVAO UNIVERSITY

MARCH 2014

**EMILIO AGUINALDO: A SINGLE PLAYER
3D ROLE PLAYING GAME USING OGRE**

An Independent Study

Presented to

The Faculty of the Computer Studies Division

Ateneo de Davao University

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

By

MICHAEL ANGELO R. CALAQUE

RONN ERIC J. LAMOSTRE

SCHOOL OF ARTS AND SCIENCES

ATENEO DE DAVAO UNIVERSITY

MARCH 2014

TABLE OF CONTENTS

	Page
Chapter 1 - Introduction	1
1.1. Background / Motivation	1
1.2. TECHNOLOGY APPLICATION CONTEXT	2
1.3. OBJECTIVES OF THE STUDY	3
1.4 SIGNIFICANCE OF THE STUDY	3
1.5 SCOPE AND LIMITATIONS OF THE STUDY	4
1.6 DEFINITION OF TERMS	4
Chapter 2 - Review of Related Literature/Work	6
Chapter 3 - PROJECT DESIGN AND METHODOLOGY	8
Chapter 4 - TECHNOLOGY BACKGROUND	10
Chapter 5 - RESULTS AND DISCUSSION	11
A. Choosing a Concept	11
B. Developing a Setting	14
B.1. Terrain	14
B.1.1. Terrain Design	14
B.1.2. Terrain Texturing	14
B.2. World Scaling	14
B.3. Exporting the Materials	14
B.4. Mesh vs. Scene Exporting	15
C. Creating a Rules of Framework	15
D. Combat, Travel and Other Actions	19
E. Character Creation	22
E.1. Character Modeling	22
E.2. Character Texturing	23
E.3. Biped Animation	25
E.4. Character-Biped Rigging	25
E.5. Mesh Animation Exporting	25
F. Populating the Game World	26
F.1. Terrain and Collision	26
F.2. Loading - User controlled first person camera and other models	26
G. Play Testing	27
Chapter 6 - CONCLUSION AND RECOMMENDATION	27
6.1. Conclusion	27
6.2. Recommendation	28

LIST OF FIGURES

		Page
1	Operational Framework	8
2	Game's start menu UI	16
3	Sample game-state pause menu UI	16
4	Exiting the game	17
5	Interacting with other characters	19
6.1.	Interacting with objects	20
6.2.	Interacting with objects	20
6.3.	Interacting with objects	21
6.4.	Interacting with objects	21
7	Sample characters	22
8	User is interacting with a character	23
9	Player is armed with a rifle	23
10	Character rendered in wireframe	24
11	Sample textured character	24
12	Character rigged with walking animation	25

EMILIO AGUINALDO: A SINGLE PLAYER 3D ROLE PLAYING GAME USING OGRE

MICHAEL ANGELO R. CALAQUE AND RONN ERIC J. LAMOSTRE, Ateneo de Davao University

The proponents had developed a game using an existing open source 3D rendering engine, game methodology, and framework. The game is based upon Emilio Aguinaldo's life during the Philippine Revolution. While the game is in full 3D, rendered using OGRE, the players are able to interact with the 3D world in First Person view. The players would be able to explore the environment and interact with the game objects and characters within the game. In creating the game's content, the 3D models; environment, characters and animation required making the game fun and visually entertaining, are designed within Autodesk's 3ds Max. Furthermore, with C++ versatility to use other libraries, the proponents were also able to include other tools and libraries in creating the game.

General Terms: Game development and design

Additional Key Words and Phrases: open source SDK, model, environment, characters, animation, 3D graphics software, image manipulation software

ACM Reference Format:

Michael Angelo R. Calaque and Ronn Eric J. Lamostre. 2013. Emilio Aguinaldo: A Single Player 3D Role Playing Game Using Ogre.

1. INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The proponents are interested in developing a game to learn the basics on how to create necessary components needed on the game such as terrain, models and characters. What are the certain processes to develop a game? How are they created? The proponents want to gain knowledge on every part of the process on developing a game especially on a three dimensional game environment. Knowledgeable of those is an advantage on the mere future and a career opportunity. Creating a game based on the life of Emilio Aguinaldo, on how he fought with the colonizers of the Philippines in acquiring our Independence. Emilio Aguinaldo a Philippine hero becoming a huge part in the war of Filipino Revolution against the Spaniards and the Americans on Post-Spanish and Pre-American time. Unlike other Filipino heroes Aguinaldo is not given much credit to what he had contributed to the Philippines primarily on History books used and taught in school and why a little detail are just known by the students. Emilio Aguinaldo gives a lot of impact on achieving the Independence of the country especially on wars against the enemies. Many Filipino heroes offered a lot and sacrifice themselves to attain our independence in many aspects as what Dr. Jose Rizal did on opening the minds of the Filipino people before against the mistreatment of Spaniard colonizers by writing novels like *Noli Me Tangere* and *El Filibusterismo*. Aside from the fact that Jose Rizal is more popular and better known Filipino hero from the students rather than Emilio Aguinaldo whom became the 1st president of the republic of the Philippines. The proponents intend to pursue this study to appreciate the life of Emilio Aguinaldo and the appreciation of the revolution that happened that resulted to the Independence of the Philippines. Philippines was under the Spanish Rule during 1521 to 1898. The Spanish quickly organized their political system upon their arrival, baptized the natives into

Christians, natives were relocated into the new settlements, and formed a government to uphold order upon its people, and because of the newly formed government, people needed to be taxed to support the colony.

The aspect in the life of Emilio Aguinaldo the proponents is focused on is how Aguinaldo achieved the Philippine Independence. The contributions made by Emilio Aguinaldo in the start of the Philippine revolution. The events in the life of Emilio Aguinaldo on the time of the Philippine Revolution. The motivation for Aguinaldo's story instead of other Filipino heroes is because Emilio Aguinaldo awakens the minds of Filipinos on the colonization of the Spaniards in the Philippines. People now a day's does not know what Emilio Aguinaldo contributed on the start of the Philippine revolution and the achievement of Philippine Independence. Where Emilio Aguinaldo became as the first president of the 1st Philippine republic. Also the building of a separate government from the colonizers to govern its own people.

Emilio Aguinaldo a Philippine hero becoming a huge part in the war of Filipino Revolution. Unlike other Filipino heroes Aguinaldo is not given much credit to what he had contributed to the Philippines primarily on many references like History books. Emilio Aguinaldo contributed a lot to the country especially on wars against the enemies. He as a hero before offered a lot and sacrifices to attain our independence. Emilio Aguinaldo has a solid vision of Philippine nationalism and where he urged the Filipinos to govern themselves and not dictated by the colonizers. His love for his country and sense of leadership to the Filipinos where proved him to be an experienced leader.

1.2 TECHNOLOGY APPLICATION CONTEXT

Implementing a 3D single player game that is less computing-intensive while using an engine that they can have more control on, the proponents need an engine that that already has OpenGL and Direct3D libraries to make the game more portable and run them to other platforms. Most gamers nowadays prefer to play three dimensional games than those in 2D wherein games have near realistic graphics. They usually prefer visually impressive games that simulate a three dimensional real world environment. Usually players like to play games that have a good storyline, gameplay, presentation and how gamers enjoy the games played. Games are goal oriented wherein missions must be accomplished to move onto a game's different levels. Gamers appreciate more the games they are playing when it has great sound ambiances and effects integrated into the game therefore emphasizes the content, draws the attention of the players, setting the mood and immersing them more. Video game developers reinforce its games with sound effects to achieve the game's reality giving the players the feel of the environment and the character they are playing making them a part of the game.

The game will be about the events happened on the life of Emilio Aguinaldo in the Philippine Revolution time. The player will be able to control the main character Emilio Aguinaldo. There are 10 levels on the game. It has a main quest wherein it includes important facts related in the story of Emilio Aguinaldo's life from the colonization period. The game is a first person view and it is an open world wherein the player can go in wherever the player wanted for a specific level on its game environment. The main character can grab guns to battle against the Spanish and American enemies. The goal of the game is to finish the entire main quest from the Spanish up to the American colonization to be able to proceed to the next game level. There are different sounds and effects on the game. Many kinds of mp3 sound format will be played and if the sound stops a playback will occur and loop the sound. Sounds effects will be generated on specific objects the player interact with and

where be heard along with the background music playing. Effects like footsteps, gunfire, wind, ammo pickup, and gun reload.

1.3 OBJECTIVES OF THE STUDY

The general objective of this study is to develop and a three dimensional role playing game with Ogre. The specific objectives are:

- To implement a single player 3D RPG based on the life of Emilio Aguinaldo during the Philippine revolution
- To design the terrain, characters, environment, sound effects and ambiences that are needed in the game of the game using 3ds Max
- To evaluate the developed game in terms of its user interface, user controls and gameplay
- To conduct user acceptance testing to 1st year history college students

1.4 SIGNIFICANCE OF THE STUDY

This study expanded the knowledge of the proponents on three dimensional game development. This includes the creation of three dimensional objects using Autodesk 3ds Max graphics software learning the concepts of polygon modeling. Applying texture mapping to characters, buildings, terrains and others that would made the game visually fun and entertaining. The used of software development kit on game development wherein it created the Ogre3d as a software package. Implemented one or more application programming interface in the form of libraries for a particular C++ programming language. The helped of sample code and supporting technical notes and documentation on starting a game development. The proponents learned about handling hardware resources like reducing the polygon count for each mesh created to render on screen, so the game will run smoothly at its optimal frame rates. In this study the proponents are knowledgeable and experienced in game design and development field which is a big advantage especially in the industries growing demand of game developers. Proficient on three dimensional animations can be a profit in the near future. In the support of the Ogre SDK, the proponents visualized and developed at three dimensional role playing game wherein the players immerse themselves and be entertained with the game.

Through the game first year college students have been able to know about the life and contributions of Emilio Aguinaldo during the Philippine Revolution. How he manages and seeks Philippine Independence that Filipinos hunger with. First year college students have been able to appreciate and learned Philippine History in an interactive way by playing the game. Facts of Emilio Aguinaldo's life are presented and other people connected with Aguinaldo which contributed to the revolution are known and recognized. Introducing the story starting from the creation of the katipunan faction Magdalo up to the achievement of Philippine independence from the americans.

Applied the Ogre3d advance ogre frameworks which provide states like menu state, game state and pause state. Those components to create a game are used and after the player has a control on playing the game whether the player wants to play or start the game, pause, resume the game, and exit the game. Loading needed headers, source class and different libraries. On every program especially on developing games there are dll's or dynamic link library file containing library functions and information's they are access by a window program. A program will not launch if the necessary dll's

needed are not with the application. It prompts that it would need those dll's to be able to start the program. In ogre sdk there are a lot of dll's needed for the game to start such as OgreMain.dll, OgreOverlay.dll, OIS.dll, Plugin_CgProgramManager.dll, Plugin_PCZSceneManager.dll, RenderSystem_Direct3D9.dll. Without those dll's the codes and develop game will not launch it sort as a reference to the program. For those who would like to research about the OGRE rendering engine, they could also use this research as their basis of their further study.

1.5 SCOPE AND LIMITATIONS OF THE STUDY

The scope of the study focused on the life of Emilio Aguinaldo during the Philippine Revolution. The focal point of the study is the appreciation of our Philippine History in other way and introducing Emilio Aguinaldo's life especially to first year students. Also the scope on game development for three dimensional single player role playing game and the game first person camera view system.

The limitations of the study focused on Ogre3D game rendering software development kit. The designed and modeling of three dimensional models, terrain, animations and mapping of textures are handled by Autodesk 3ds Max. The game's framework used is the Ogre3d advance ogre framework. Applied Ogre3D minimal ogre collision detection library. The Irrklang sound library for the game sounds, audio, ambience and effects. Directshow texture system for playing video cut scenes on each quests.

1.6 DEFINITION OF TERMS

OGRE (Object-Oriented Graphics Rendering Engine)

- Is a scene-oriented, flexible 3D engine written in C++ designed to make it easier and more intuitive for developers to produce applications utilizing hardware-accelerated 3D graphics. The class library abstracts all the details of using the underlying system libraries like Direct3D and OpenGL and provides an interface based on world objects and other intuitive classes.

Scene Manager

- Everything that appears on the screen is managed by the Scene Manager. When you place objects in the scene, the Scene Manager is the class which keeps track of their locations. When you create Cameras to view the scene the Scene Manager keeps track of them. When you create planes, billboards, lights...and so on, the Scene Manager keeps track of them.

Direct3D

- An API for manipulating and displaying three-dimensional objects. Developed by Microsoft, Direct3D provides programmers with a way to develop 3-D programs that can utilize whatever graphics acceleration device is installed in the machine. Virtually all 3-D accelerator cards for PCs support Direct3D.

Minimal Ogre Collision

- MOC (Minimal Ogre3d Collision) is a small library with the aim to provide simple and easy to use collision and mouse-picking functionality for the Ogre3D graphics engine written by Thomas Gradl (Nauk) and Esa Kylli (wannabee). It is based on the Ogre3D Wiki entry "Ray casting down to Polygon-Level".

IrrKlang

- IrrKlang is a high level 2D and 3D cross platform (Windows, Mac OS X, Linux) sound engine and audio library which plays WAV, MP3, OGG, FLAC, MOD, XM, IT, S3M and

more file formats, and is usable in C++ and all .NET languages (C#, VisualBasic.NET, etc).

Directshow

- This plug-in has been developed using Directshow libraries, that's why is only available for Windows platforms. This plug-in allows using a video file as texture for a material, being able to play near any file that can be played in Windows Media Player.

Advance Ogre Framework

- An out – of – the – box running advanced framework of an Ogre application offering Ogre initialization, game/application state framework, custom render loop, Dot Scene / XML scene loading, different keyboard input modes, user interface, in-code material manipulations, ray scene queries with query masks.

Autodesk 3ds Max

- A 3D computer graphics software for making 3D animations, models, and images. It was developed and produced by Autodesk Media and Entertainment. It has modeling capabilities, a flexible plug-in architecture and can be used on the Microsoft Windows platform.

2. REVIEW OF RELATED LITERATURE AND WORKS

In developing 3D games, things needed are; a 3D graphics software, audio editing software, a game engine, and a mid or high end desktop computer to develop the game on, and most importantly some concept art and storyboarding to have the game based upon. Large gaming industries purchase leading 3D game engines and graphic software that provides powerful tools that enables the developers to focus more on creativity than technical challenges. These concepts and tools determines what type of game that will be developing and where mastering is a huge step to make a three dimensional game.

Though the proponents' thesis will be about game development and design, the review and research materials would be focusing more on the game's components and what libraries should, ideally, be used in the game. The proponents are looking into some notable and emerging game engines, comparing them while reviewing a summary of how they work. Basically, a game engine provides a software framework, wherein it already has a rendering engine, physics engine, sound, scripting, animation, artificial intelligence, networking, and memory management. Almost all the game engines already available have the same features, but how do they all differ depends on what libraries and improvements they used in their engine. Currently, the most talked about game engine used is the CryENGINE. Some games that are associated with CryENGINE are the Far Cry series, and Crysis series, both of which made a success and were critically acclaimed, gained numerous awards, and well-reviewed by gamers with no disappointments. With great enthusiasm, the proponents found out that good storylines are not enough; it is the realism that game developers add to their games adds a competitive edge over other game titles. CryENGINE boasts having the fastest high-end renderer in the world, now has DirectX 11 support, the future of game development, wherein it has enhanced game HDR visuals, improved physics and performance. Game artists uses HDR a lot these days, it gives the environment some dynamics, feel and most importantly illumination.

Currently, another emerging game engine that is constantly being improved and developed on is the Unity Engine. Unity Engine assures its developers that their engine is the ultimate tool for video game development, and architectural visualizations. Its tools provide reliable performance, optimized graphics, and offers multiplatform capabilities. Games like Temple Run, Shadowgun, and the most talked about first person horror game, Slender: The Arrival wouldn't be possible without the Unity Engine.

This research gets tricky because the game engines reviewed seem to be all the same. The game engines researched, almost have the same features, concepts, and goals to produce a game. Average gamers don't even notice much difference when playing the game. Gamers just say how good the graphics were, the special effects, the animation and how realistic were they rendered. Basing everything on graphics in a standpoint is difficult. The proponents found out, not all game engines are the same. They all differ in how they perform large calculations just to render a scene. Certainly, from time to time, some parts of the engine should be improved to efficiency, maximize quality that also reduces bottlenecks. In this thesis proposal, the proponents want to develop a game, that the engine used can easily be improved in which the engine itself can use and import other libraries to develop the game, which the other game engines are incapable of. Game engine such as RAGE, previously mentioned in the review, is proprietary and closed source, modifying its source may cost the developers.

In order to create the necessary characters, terrain and objects that are needed for the game, the proponents used Autodesk 3ds Max to create those. 3ds Max is much more stable and the probability of encountering software bugs during the content creation process is low. Furthermore, the

proponents used 3ds Max to create the content because it has lots of support and plugins found online. In using 3ds Max, the proponents can choose modifying the models between polygon or in mesh mode. Modifying models in polygon or in mesh mode have its own advantages to achieve the desired model's result. Also, the process of animating in 3ds Max is much more faster since the software also provides the biped system to easily rig the models. What is more important in the content design process is the proper scaling of its objects relative to its real world scale. Fortunately, 3ds Max has this feature and is very useful to the proponents while creating the 3d models. 3ds Max has its own built in units whether in Metric or US standard of measurements. In creating the models, the proponents have been switching between meters, feet and inches to achieve the right scales. Having to add more detail to the 3D environment, the proponents were also able to apply texture mapping to the 3d objects created. The proponents were able to apply the textures using 3ds max's Material Editor. With the material editor's features, the proponents were able to apply diffuse bitmap-maps, vertex-painting, and displacement map.

The proponents did use Ogre3D rendering engine. It is written in C++ to make it easier and intuitive for developers to produce applications utilizing hardware-accelerated three dimensional graphics especially on game development. In the Ogre official website, classes under the Ogre SDK are discussed. On each class different functions are given comments to give developers an idea on the purposes of functions and the parameters needed. It also present available libraries which are supported by the Ogre3d platform like CEGUI, MOGRE, MyGUI, OgreOggSound, OIS, Caelum, PagedGeometry Engine, Particle Universe plug-in, Berkelium, OgreBullet, OgrePhysX. Those libraries can be imported and applied to game development. There are frameworks provided and updated by the Ogre community. Concerns on Ogre and c++ programming is catered by admins and developers on the forum. As they give suggestions and comments to improve more on users projects. Those frameworks are classified to whether the framework is applicable to a small or large project. There are tutorials given by Ogre to start up a project. The Ogre3d forums which is subdivided into different topics such as Announcement/News/Information, General Discussion, Help, Featured Request, Using OGRE in practice, Artists & Content Creators, Back to Basics, Showcase, Events & Meetings, Lounge/Off-topic and Add-ons. There are active users to help developers in starting a project. These users separately contribute to fix bugs and errors in every update of libraries supporting Ogre. It is more flexible in loading models with dot mesh extensions created from 3ds Max which is also supported by Ogre. The models can be positioned in x, y and z coordinates. On Game Engines developers apply pre-rendered models that are provided and link it to the scene while in Ogre objects are loaded detail by detail. Differ The developer has a wide range using those tools to enhance more the development. Different models can be attached as one. Example of it is attaching a gun entity to the human entity. Ogre can specify where to attach the entity basing on the skeleton of each model. Those can be rotate in yaw, pitch and roll. The camera system of Ogre can be manipulated in first person view, first person view and free look. Gathering of meshes into one scene created in 3ds Max can be loaded directly on Ogre scene manager.

For the proponents, they want an engine that is design-led rather than feature-led and favors design quality. The proponents also want flexibility in terms of game development, wherein other libraries and techniques can be used. The OGRE 3D Engine fits the description of what the proponents are looking for. Community pages of the OGRE Engine provide a lot of libraries and a lot of freedom for the developers to choose from. This way, developers have way more control over the libraries used. Multiple libraries, ranging from terrain, physics, collision, HDRlib to GUIs are available and constantly being improved by contributing open source developers.

3. PROJECT DESIGN AND METHODOLOGY

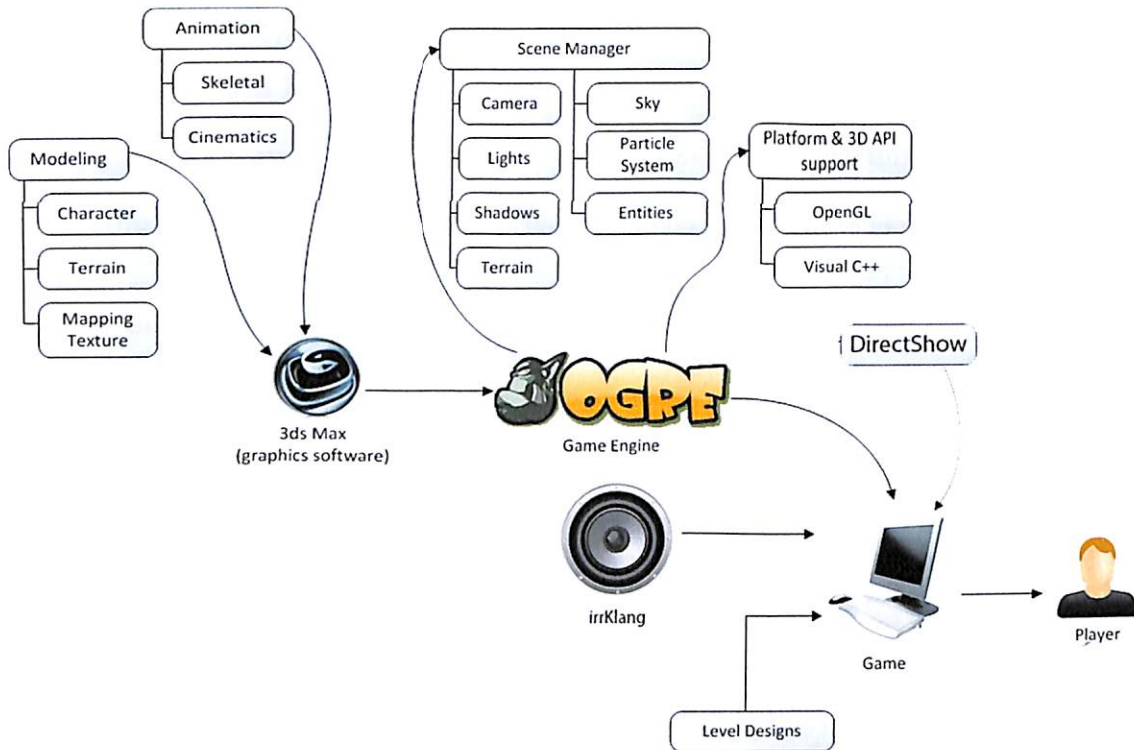


Figure 1. Operational Framework

Autodesk 3ds Max is used on creating the different models, such as the characters, terrain, texture mapping, and skeletal animation. The 3D game rendering engine OGRE is which includes a software development kit. It has a built in scene manager that handles camera, lights, shadows, terrain, sky, particle system and entities. OGRE already provides OpenGL or DirectX which are the default renderer for the game. For the sound, audio and effect, the proponents implemented the IrrKlang sound library to make the game more interactive. While the Directshow texture system for playing video cut scenes in the game.

An overview of role-playing game development by J. Edward Casteale

A. Choosing a Concept

The proponents came up with a concept of a role playing game where the player controls only one character. The player can travel different levels after finishing the main quest. Associating the game with Philippine history relating to a life of a Filipino hero. The proponents choose General Emilio Aguinaldo as the main character for the game.

B. Developing a Setting

Creation of terrains, houses and other entities from Autodesk 3ds Max. The game world is linked to the Philippine revolution times. The proponents created small towns where there are a just limited houses and objects to interact with the player. The player can interact to specific

characters in the game. There are moving characters walking to specific areas. Objects like guns can be interacted by the player.

C. Creating a Rules of Framework

The proponents used the Advance Ogre Framework consisting of 17 classes. On this framework there are three states which are the menu state, pause state, and game state. The proponents apply graphical user interface on each states. In the menu state there are gui's for main menu which is the start of the game. There are two button on it the start game and the exit. In the pause state there are three gui's return to game, return to main menu and exit, while in the game state there separate functions to create the gui's for each level.

D. Combat, Travel and Other Actions

After creating the concept and a setting for the game, the proponents were able to develop the general framework for the rules that the game is running on. The proponents were able to discuss with themselves, foreseeing the character's movements, and interaction with the other characters in the game. The proponents have chosen to make use of the other games usual controls and somewhat standard controls. Performing character movement for the first person camera is done in keyboard keys. The proponents applied the usual keyboard keys used in role playing games and other types of games. The movement of the camera for the x axis is performed in the mouse. The mouse clicks handles the interactions of the character to the objects. There is provided crosshair for the attached object to the camera.

E. Character Creation

The characters that are needed for this game are based from the quests that are laid out from the story. Some of the characters that are modeled are Baldomero Aguinaldo, Andres Bonifacio and feature other soldiers that is needed for the war scene. Characters are modeled inside Autodesk's 3ds Max. 3ds Max has all the tools and plugins making it possible to make the necessary models.

F. Populating the Game World

In making the game world feel authentic, the game is being populated with foliage, textures, and models that are seen within the story's timeline.

G. Play Testing

The game testing is conducted to 1st year history college students. They had evaluated the game's story and how it is depicted. Moreover, they also tested the game's user interface, user controls and overall gameplay.

4. TECHNOLOGY BACKGROUND

Autodesk 3ds Max is a computer graphics software for making 3D animations, models and image. Autodesk 3ds Max can create different models that the proponents can apply on the game such as houses, guns, and skeleton characters and with the use of a modeling technique like polygon modeling that models created have an exact standard measurement how small or big an object is. Skeletons can be generated in the Autodesk 3ds Max and skinning of it afterwards creates an individual character that can be used in the game and with the animation applied on each created moving models will make the exquisite. It also generates scenes where different entities, objects and terrain are loaded inside. Those are exported through Autodesk 3ds Max that releases different formats along with the dot scene such as dot material for the xml script, dot mesh and dot skeleton for the animation movement setting up fps or frame rate per second on the skeleton movement. Lastly, is the texture with any kind of image format like png and jpeg. The three exported files are in the dot scene which will be loaded in the game. Also these files must be place in the Ogre sdk media folder. The material files are placed in the material folder. The mesh and skeleton file place in the models folder while the scene file in a specific folder in which the path is set in the ogre's resources dot cfg format. OGRE3D is a graphics rendering engine to produce applications utilizing hardware-accelerated 3D graphics. OGRE is a fully multi-platform with Direct3D and OpenGL support. Can be used to provide an interface based on world objects and other high level classes.

OGRE has an object-oriented design with plug-in architecture that allows easy addition of different features thus making it highly modular. It can render the same content on different platforms without the content creator having to take into consideration the different capabilities of each platform. This reduces the complexity of deploying a game on multiple systems. Pre-compiled binaries exist for Linux, Mac OS X, and all major versions of Windows. The animation engine has full support for hardware weighted multiple bones skinning, which can be fixed across several poses for full pose mixing. The libraries also feature memory debugging and loading resources from archives. Content exporter tools available for 3D modelers including Autodesk 3ds Max. The Ogre sdk also have a cross platform library that supports human interface devices like keyboards, mice, and joysticks. OIS is an object oriented input system so that the game will incorporate with those output devices. IrrKlang is a cross platform sound library for c++, c# and all .net languages. irrKlang is free for non commercial use . irrKlang is a powerful high level API for playing back sound in 3D and 2D applications like games, scientific visualizations and multimedia applications. It supports many kinds of music formats such as wav, ogg, mp3, flac, mod, it, s3d, xm. irrKlang has built-in support for 3D sound on all platforms and audio drivers. It is designed to use in 3D games. It is very efficient and does not use a lot of processing time. IrrKlang is able to play short buffered sounds like gun shots, streamed sounds in 3D. A fast 3D audio on low end hardware. Some low end audio hardware is not able to play 3D sounds without significant performance loss or some do not support 3d audio buffers. irrKlang has a high performance 3d sound buffer emulator built-in, causing a sound experience nearly as it was a real 3d sound buffer. Multiple roll off models, irrKlang supports linear as well as logarithmic roll off.

5. RESULTS AND DISCUSSION

The proponents followed J. Edward Castele's methodology on his take of an overview of role-playing game development. As Castele said in his article, creating an RPG game considers a lot of preparation and planning, and takes a lot of dedication. On his take of the overview of role-playing game development, the proponents followed each step to achieve the desired goal of the development. His overview starts with choosing a concept, and that it is, the proponents based the game from the life of Emilio Aguinaldo into a 3D RPG game. Wherein players can immerse themselves into it and the proponents can incorporate fun in learning, on appreciating Aguinaldo's story and his contributions in our Philippine history put into the game.

A. Choosing a Concept

The concept of the game will be based upon the life of Emilio Aguinaldo, from those written into history books, the proponents will be able to create the different quests that the players will go through in the game. The game's quests are as follows:

Based on Story:	Game Quests:
In 1894, Aguinaldo joined the Katipunan or the K.K.K., a secret organization led by Andrés Bonifacio, dedicated to the expulsion of the Spanish and independence of the Philippines through armed force. Aguinaldo used the nom de guerre Magdalo, in honor of Mary Magdalene. His local chapter of the Katipunan, headed by his cousin Baldomero Aguinaldo, was also called Magdalo.	Quest #1 - Organizing the K.K.K - (MAGDALO) 1.) Find the painting of Maria Magdalena in the walls 2.) Go find your cousin baldomero aguinaldo in the area 3.) Talk with baldomero aguinaldo about organizing the magdalo 4.) Argue with baldomero aguinaldo about creating an illegal organization (MAGDALO) and answer his questions
By December, the Spanish authorities in Manila recognized three major centers of rebellion: Cavite (under Emilio Aguinaldo and others), Bulacan (under Mariano Llanera) and Morong (now part of Rizal, under Bonifacio). Bonifacio served as tactician for the rebel guerillas though his prestige suffered when he lost battles he personally led.	Quest #2 - The Start of the Revolution 1.) Talk with the armies about the revolution that will happen 2.) Grab necessary guns and ammo's in preparing for the Filipino revolution 3.) Go back to the kuta(base) to plan the attacks
On 17 February 1897 Aguinaldo and a group of katipuneros defeated Spanish forces led by General Camilo de Polavieja at the Battle of Zapote Bridge in Cavite. General Edilberto Evangelista, civil engineer, revolutionary and trench	Quest #3 - Battle of Zapote 1.) Defeat all the enemy armies 2.) Infiltrate the enemy bases 3.) Kill Gobernador-Heneral Camilo de Polavieja

<p>builder, was killed in the battle. The province of Cavite gradually emerged as the Revolution's hotbed, and the Aguinaldo-led katipuneros had a string of victories there.</p>	
<p>Conflict between the Magdalo and Magdiwang Katipunan factions led to Bonifacio's intervention in the province of Cavite. The Cavite rebels then made overtures about establishing a revolutionary government in place of the Katipunan. Though Bonifacio already considered the Katipunan to be a government, he acquiesced and presided over a convention held on 22 March 1897 in Tejeros, Cavite. There, the republic of the Philippines was proclaimed, with Aguinaldo being elected president.</p>	<p>Quest #4 - Magdalo-Magdiwang Conflict</p> <p>1.) Locate Andres Bonifacio kuta(Base)</p> <p>2.) Interact and dialogue with Andres Bonifacio about the</p> <p>3.) Capture Andres Bonifacio together with his siblings about the false accusation made pertaining to Magdalo</p>
<p>At Aguinaldo's orders, Bonifacio and his brothers were arrested and, in a mock trial lasting one day, convicted of treason, and sentenced to death.</p>	<p>Quest #5 - The Mock Trial</p> <p>1.) Talk with the judges on the trial of Andres Bonifacio</p> <p>2.) Find Gen. Manuel Noriel</p> <p>3.) Talk with him about the trial</p>
<p>After some vacillation, Aguinaldo initially commuted the death sentence, but cancelled his commutation order after being convinced by General Manuel Noriel, president of the Council of War, and others prominent in his government that the death sentence must stand. Andrés and Procopio were executed by firing squad on 10 May 1897 at Mount Hulog, Maragondon, Cavite.</p>	<p>Quest #6 - Execution of Andres and Procorpio</p> <p>1.) Go to the mountains of Hulog</p> <p>2.) Interact with the people to be guided on the exact location</p> <p>3.) Find the firing squad that will execute the brothers</p>
<p>From as early as March 1897, Fernando Primo de Rivera, as Spanish Governor-General of the Philippines had been encouraging prominent Filipinos to contact Aguinaldo for a peaceful settlement of the conflict.</p>	<p>Quest #7 - Biak-na-Bato Agreement</p> <p>1.) Find the person about the agreement of Biak-na-Bato</p> <p>2.) Find the paper in the table to put in the required details</p> <p>3.) Enter the value of the payment stated in the agreement</p>
<p>On 9 August, Manila lawyer Pedro Paterno met with Aguinaldo at Biak-na-Bato with a proposal for peace based on reforms and</p>	<p>Quest #8 - Surrendering the government</p> <p>1.) Find Pedro Paterno and other important people in the port</p> <p>2.) Talk with Pedro Paterno about the payment instalment on the agreement of Biak-na-Bato</p>

<p>amnesty. In succeeding months, Paterno conducted shuttle diplomacy, acting as an intermediary between de Rivera and Aguinaldo. On 14 December and 15 December 1897 Aguinaldo signed the Pact of Biak-na-Bato, under which Aguinaldo effectively agreed to end hostilities and dissolve his government in exchange for amnesty and "\$800,000 (Mexican)" (Aguinaldo's description of the amount) as an indemnity.</p>	<p>3.) Find the ship named "Uranus" 4.) Find the captain of the Uranus ship 5.) Interact with the captain to stop by on Hong Kong 6.) Find Pedro Paterno in the ship 7.) Get the money to Pedro Paterno for the first installment payment 8.) Go to the bank in Hong Kong to deposit the \$400,000 9.) Interact with the person inside the bank to give the money</p>
<p>On April 25, the Spanish–American War began. While the war mostly focused on Cuba, the U.S. Navy did have an Asiatic Squadron in Hong Kong and commanded Commodore George Dewey. It sailed for the Philippines and on 1 May 1898, in the Battle of Manila Bay, engaged and destroyed the Spanish Pacific Squadron, and proceeded to blockade Manila. Several days later, Dewey provided transportation to Aguinaldo from Hong Kong to the Philippines. Aguinaldo promptly resumed command of revolutionary forces and besieged Manila. On 24 May 1898 in Cavite, Aguinaldo issued a proclamation in which he assumed command of all Philippine forces and established a dictatorial government with himself as dictator.</p>	<p>Quest #9 - The return of Aguinaldo 1.) Find the person in the port of Hong Kong 2.) Talk with the person about the war of Spain-America in the Philippines 3.) Find the ship to board back to the Philippines 4.) Find the captain in the ship 5.) Talk to the captain to let you return to the Philippines</p>
<p>On 18 June, after issuing the Philippine Declaration of Independence from Spain on 12 June, Aguinaldo issued a decree formally establishing his dictatorial government. On 23 June, Aguinaldo issued a decree replacing his dictatorial government with a revolutionary government, with himself as President.</p>	<p>Quest #10 - The Philippine Independence 1.) Find the army general in the area 2.) Talk with the army general about declaring the Philippine government 3.) Find the residential house of Aguinaldo's in Cavite el Viejo 4.) Find Ambrosio Rianzares Bautista 5.) Get the documents on her about the laws on declaring the Philippine independence 6.) Go out to the house to give the message on the declaration stated in the document and as the 1st president of the Philippine republic</p>

B. Developing a Setting

In developing the game's setting, the proponents should stick to the story's theme and era. From there the objects in the game should look and feel as if it is within that timeline. There are only a few images and illustrations available in books and internet resources about; for example, Emilio Aguinaldo's house and its surrounding environment. Though that's the case, we tried to imitate of what is only given from those resources; the houses, ground, sky and level of textures that are going to be applied to the 3D models. Since the game is in 3D, we made it look as close as possible to achieve the desired look of the game.

B.1. Terrain

B.1.1. Terrain Design

The VertexPaint Modifier is used to design the terrain. VertexPaint is able to create landscapes, mountains or specifically heights in map. To add heights in map, the plane applied should have a number of segments to avoid a jagged look for the terrain. The vertex paint modifier, by default, is set to "disable vertex color display" which prevents from seeing what is going to be painted. Using the buttons from the top of the window, set the mode to "vertex color display - unshaded" which displays only the vertex color on the model, regardless of lighting, object color or materials. In this mode the effect of the paint can be clearly seen in the vertices. Before starting to paint, switch to top view and adjust the z-elevation of the model in order to get it just in the middle of the 3d-grid. That way, the grid can be seen passing through the terrain and use it as a reference for the highs and lows of the terrain. We are going to paint with the brush using black. The opacity parameter can be lowered in order to get more blur between the white and black.

B.1.2. Terrain Texturing

The VertexPaint Modifier has a Displacement Modifier applied. Using the Displacement master material, select a texture that is going to be applied to the terrain through the diffuse map.

B.2. World Scaling

3ds Max keeps track of all measurements in its own internal system unit. No matter what kind of display units being used, measurements are stored in this absolute unit for storage and computation. The default system unit is defined as 1.000 inch. As long as the system unit is left at one inch, you can freely share models and change units on the fly with no effect on the underlying geometry. This means merging a model can be created with any standard unit into your scene at true scale. By using the unit measurements, the objects rendered in the scene are properly scaled. When you change display units, 3ds Max displays measurements in the new unit for your convenience. All dimensions are displayed in the new unit. Essentially, you're using a new "measuring stick." No object is changed in this process. As in the physical world, objects in the scene maintain their absolute size, regardless of how you measure them.

B.3. Exporting the Materials

Materials can be exported using the OgreMax Plugin. In the Scene Settings, the "Export Materials to Separate Files" has to be checked, under the Materials tab. Also, the "Include Parent Material in Name" should be checked to avoid export issues.

B.4. Mesh vs. Scene Exporting

A scene is different from a mesh. A mesh should only contain as a single entity. A Scene file should only be used to export a file that contains the terrain with multiple objects in it. The scene file is helpful when importing the terrain with DotSceneLoader.

The models have to be exported so the models can now be used within Ogre SDK. To export the models, characters and its terrain, the proponents used the OgreMax plugin provided by the Ogre SDK's website.

C. Creating a Rules of Framework

There are many kinds of framework provided by ogre3d such as ogre wiki tutorial, minimal ogre, tiny ogre, low level ogre and advance ogre. The proponents use the advance ogre framework containing 17 classes. The AppStateManager class loads different states. Also under the framework are states like game state, pause state and menu state. The proponents also used other frameworks but those are applied to small projects and for tutorial demos. Unlike the advance ogre framework where in used for large and commercial projects. The framework provides the needed development for any kind of projects. Advance ogre framework has a game state system, a graphical user interface, different input modes, scene loading, and manual material manipulation. Game state system handles events directly and update on the players screen and it is like a separate program within the game. Those states are menu state, game state and pause state. Those states have functions like enter, createscene, update, and exit. Menu state is the first state that comes first which displays the graphical user interface (GUI) of the game. The proponents use button widgets with captions of start game and exit. Inside the buttonHit method there is an if-else condition.

```
void MenuState::buttonHit(OgreBites::Button *button)
{
    if(button->getName() == "ExitBtn")
        m_bQuit = true;
    else if(button->getName() == "EnterBtn")
        changeAppState(findByName("GameState"));
}
```

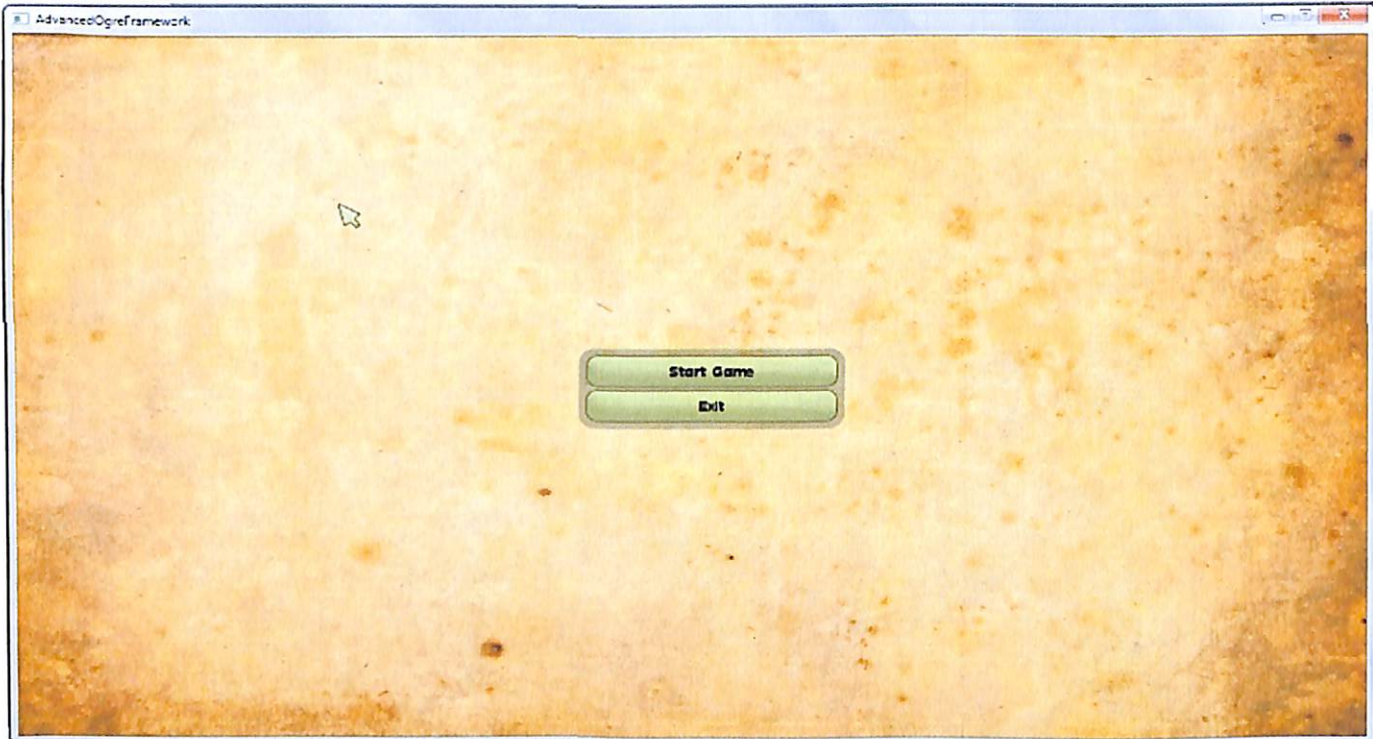


Figure 2. Game's start menu UI



Figure 3. Sample game-state pause menu UI

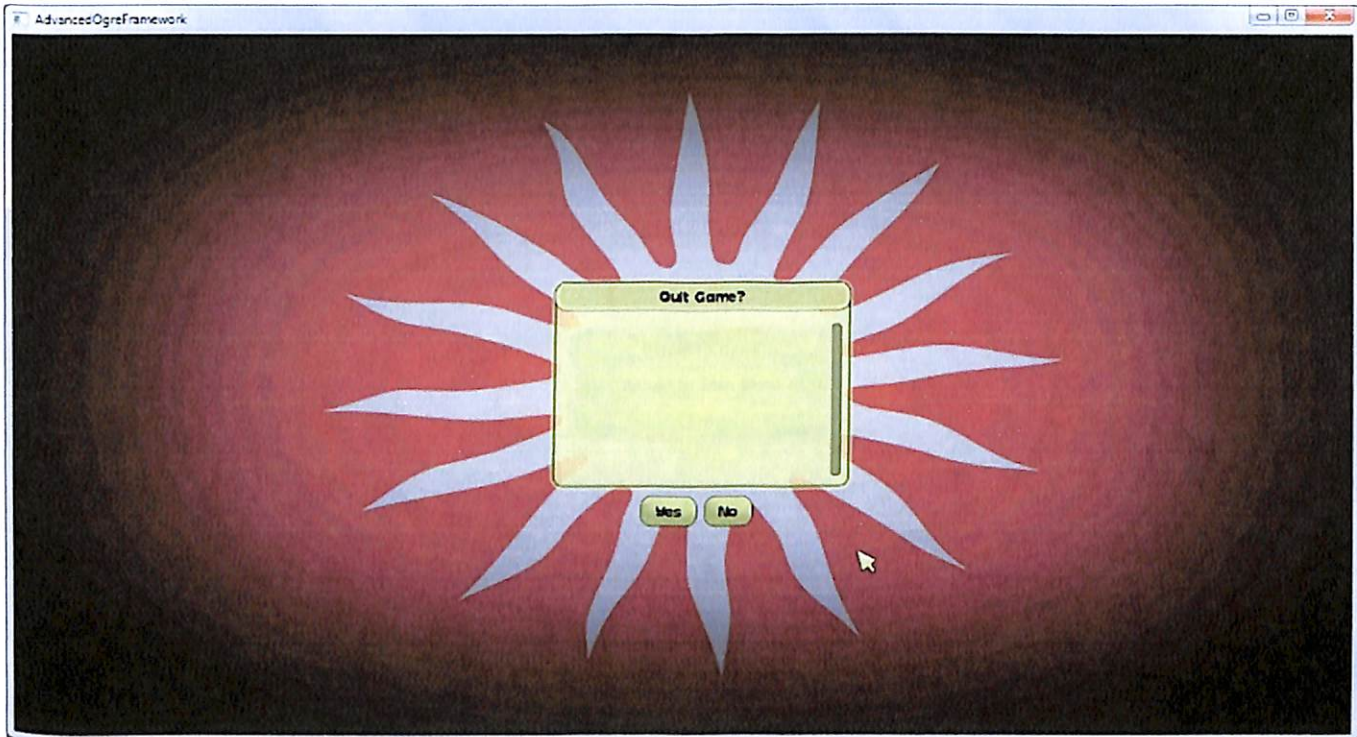


Figure 4. Exiting the game

If the button “EnterBtn” is hit then it launch the game state while if the button “ExitBtn” is hit then it exits the current state. Texture background is used to display the image of Emilio Aguinaldo in the menu screen. In this state after the start button is clicked the menu state is destroyed while loading the next state the state the game state. Creating and setting up a background for the menu screen. The code for creating the menu screen background material.

```
Ogre::MaterialPtr material = Ogre::MaterialManager::getSingleton().create("Background2",
"General");
material->getTechnique(0)->getPass(0)->createTextureUnitState("aguinaldo4.jpg");
material->getTechnique(0)->getPass(0)->setDepthCheckEnabled(false);
material->getTechnique(0)->getPass(0)->setDepthWriteEnabled(false);
material->getTechnique(0)->getPass(0)->setLightingEnabled(false);
```

The image naming aguinaldo4.jpg from the ogre sdk resource under the media folder that is set from the project application resources.cfg. Creating a 2d rectangle so that the material created will be attached to.

```
Rectangle2D* m_Rect = new Ogre::Rectangle2D(true);
m_Rect->setCorners(-1.0, 1.0, 1.0, -1.0);
m_Rect->setMaterial("Background2");
```

Lastly, attaching the node in the scene manager on the rectangle created named Background2 to be able to view the material. Pause state is where current state of the game can be pause and resumed. There is also a buttonHit method like from the menu state class.

```
void PauseState::buttonHit(OgreBites::Button *button)
```

```

{
    if(button->getName() == "ExitBtn")
    {
        OgreFramework::getSingletonPtr()->m_pTrayMgr->showYesNoDialog("Sure?",
"Really leave?");
        m_bQuestionActive = true;
    }
    else if(button->getName() == "BackToGameBtn")
        m_bQuit = true;
    else if(button->getName() == "BackToMenuBtn")
        popAllAndPushAppState(findByName("MenuState"));
}

```

Exiting the game state and resuming the current state. Lastly, is the Game state loads necessary object or entity in the game and also where the game is played. Under this state functions of enter, createscene, exit, pause, resume, movecamera, getinput, buildgui and update. Also there are key events and mouse events for the game state like keypressed, keyreleased, mousmoved, mousepressed, mousereleased and onleftpressed. Ogre Sdk trays are use to build the game gui to display on the players screen. Sdk trays have widgets to apply such as button, textbox, select menu, label, separator, slider, params panel, checkbox, decor widget, progress bar and null tray. The sdk tray is composed of a header and a class that is provided in the ogre sdk. The design of each widget is under the sdk trays zip file in the ogre sdk media. Inside the game state there is a function buildgui for creating the sdk trays widgets.

```

void GameState::buildGUI()
{
    OgreFramework::getSingletonPtr()->m_pTrayMgr-
>showFrameStats(OgreBites::TL_BOTTOMLEFT);
    OgreFramework::getSingletonPtr()->m_pTrayMgr-
>showLogo(OgreBites::TL_BOTTOMRIGHT);
    OgreFramework::getSingletonPtr()->m_pTrayMgr->createLabel(OgreBites::TL_TOP,
"GameLbl", "Game mode", 250);
    OgreFramework::getSingletonPtr()->m_pTrayMgr->showCursor();
    Ogre::StringVector items;
    items.push_back("cam.pX");
    items.push_back("cam.pY");
    items.push_back("cam.pZ");
    items.push_back("cam.oW");
    items.push_back("cam.oX");
    items.push_back("cam.oY");
    items.push_back("cam.oZ");
    items.push_back("Mode");

    m_pDetailsPanel = OgreFramework::getSingletonPtr()->m_pTrayMgr-
>createParamsPanel(OgreBites::TL_TOPLEFT, "DetailsPanel", 200, items);
    m_pDetailsPanel->show();
}

```

The tray manager holds the created widgets where it can position it.

D. Combat, Travel and Other Actions

The usual movement controls for games the keyboards keys W for forward, S for backward, A to move left and D to move right. The ESC key is use to exit the game if it is on menu mode. Then another ESC key is for pausing the game and graphical user interface pops out and choose to the buttons, return to game, return to main menu and exit game. Interact with character entities is by clicking the left mouse. The key ENTER is for skipping or stopping the interaction audio from the character entity. Keyboard key G for grabbing an object like guns. After grabbing a gun there is an automatic circular crosshair for the attached gun. When the gun is grabbed and present on the player's screen mouse left click is for firing bullets. Right click for mouse is for gun sight to be able to zoom and fire a target. Ctrl button is for crouch for the player to hide from enemy bullets.

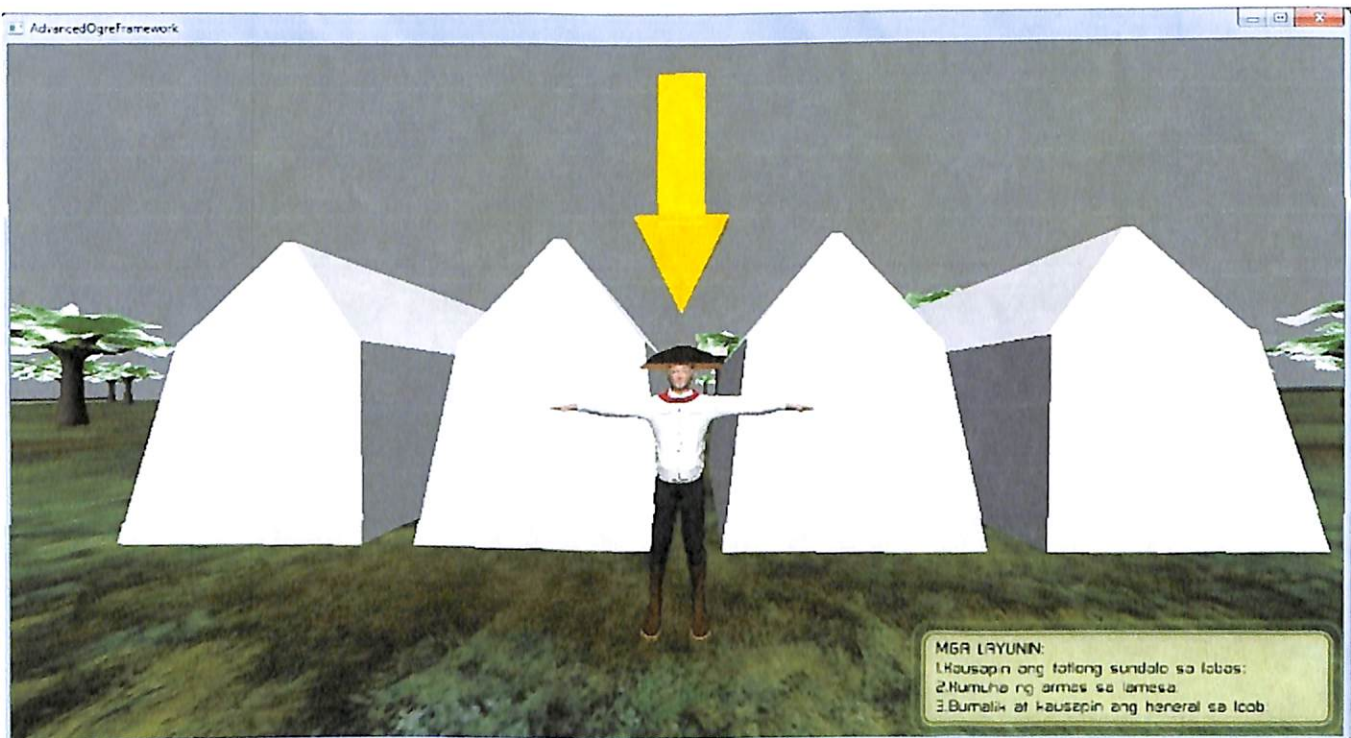


Figure 5. Interacting with other characters

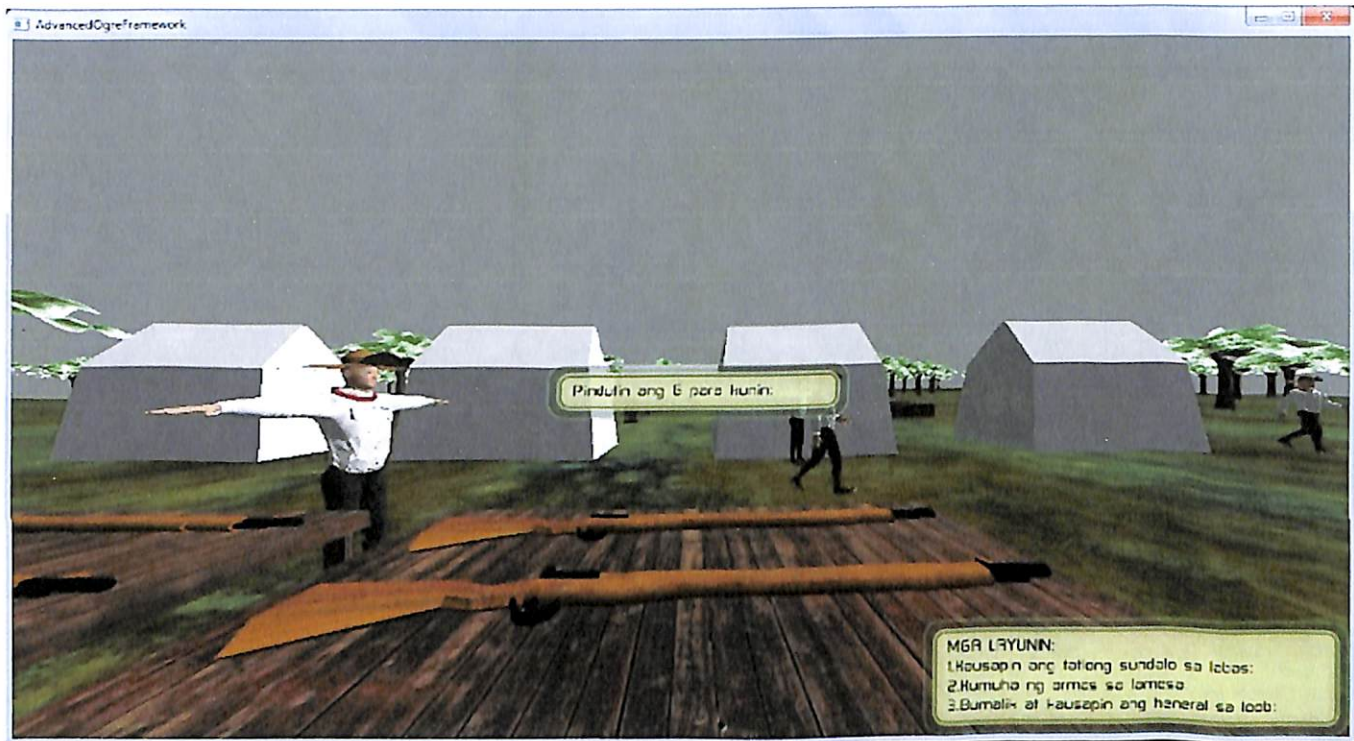


Figure 6.1. Interacting with objects

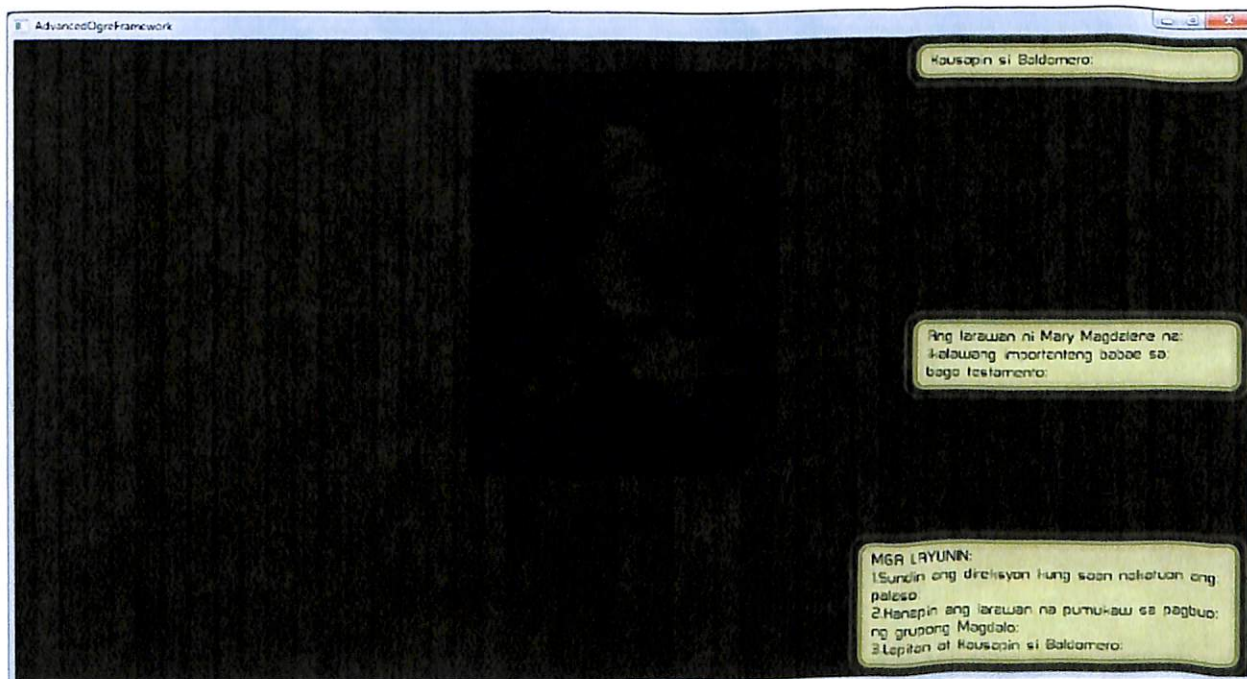


Figure 6.2. Interacting with objects

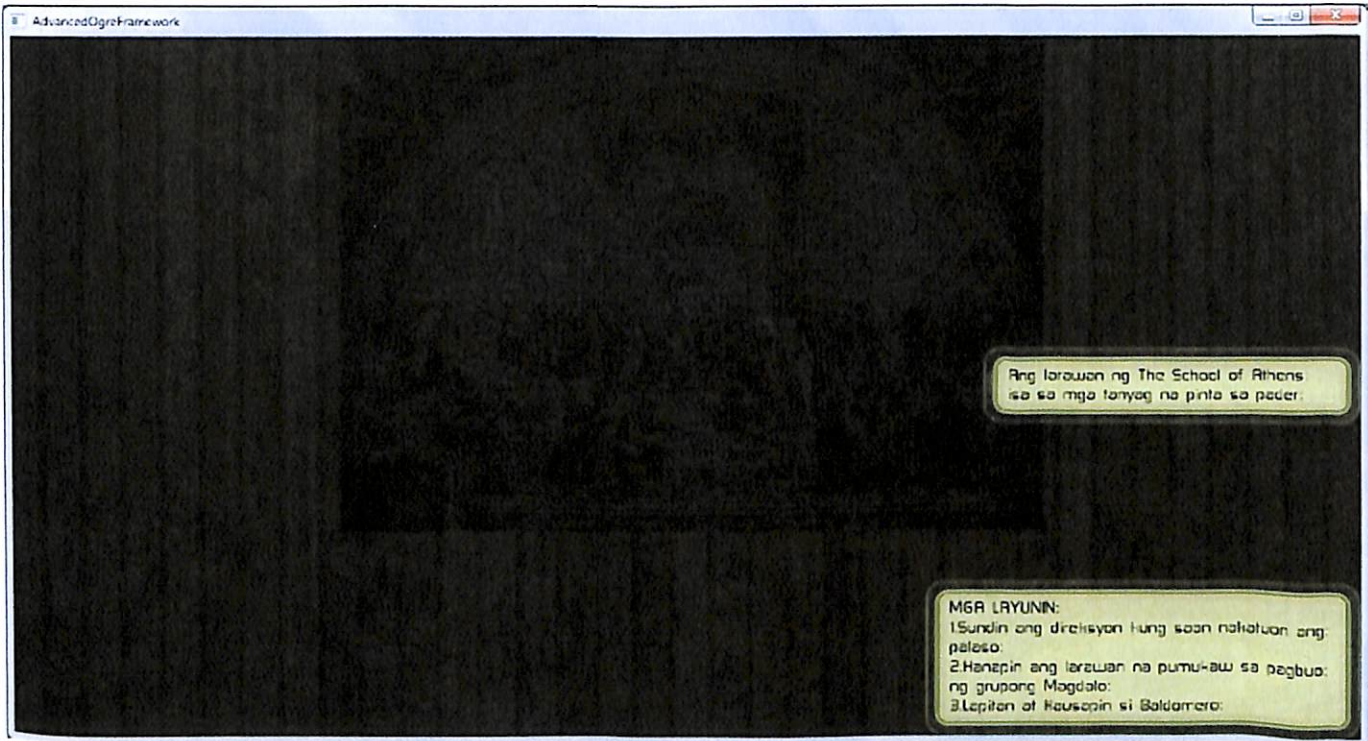


Figure 6.3. Interacting with objects

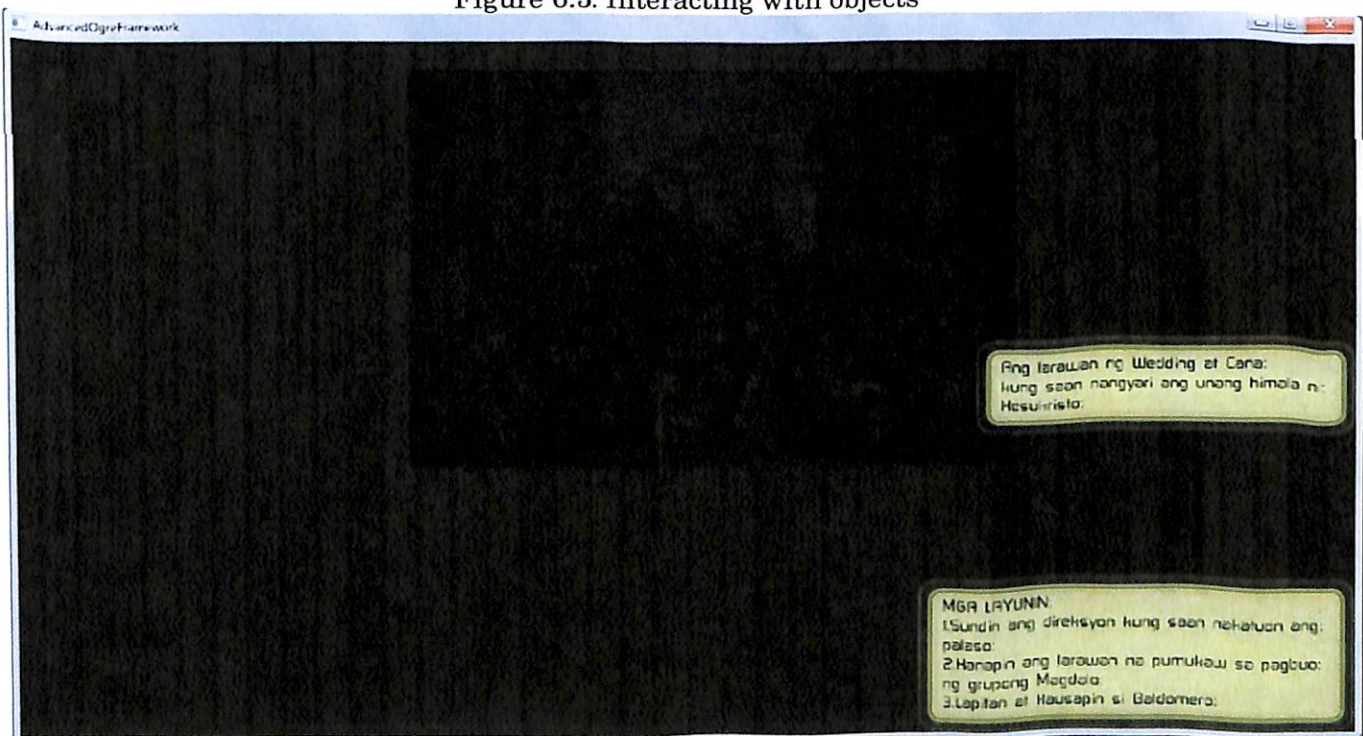


Figure 6.4. Interacting with objects

E. Character Creation

The 3D models have been designed and created within 3DS Max. For this section, the 3D content design process will be divided into sections so everything will be discussed thoroughly below.

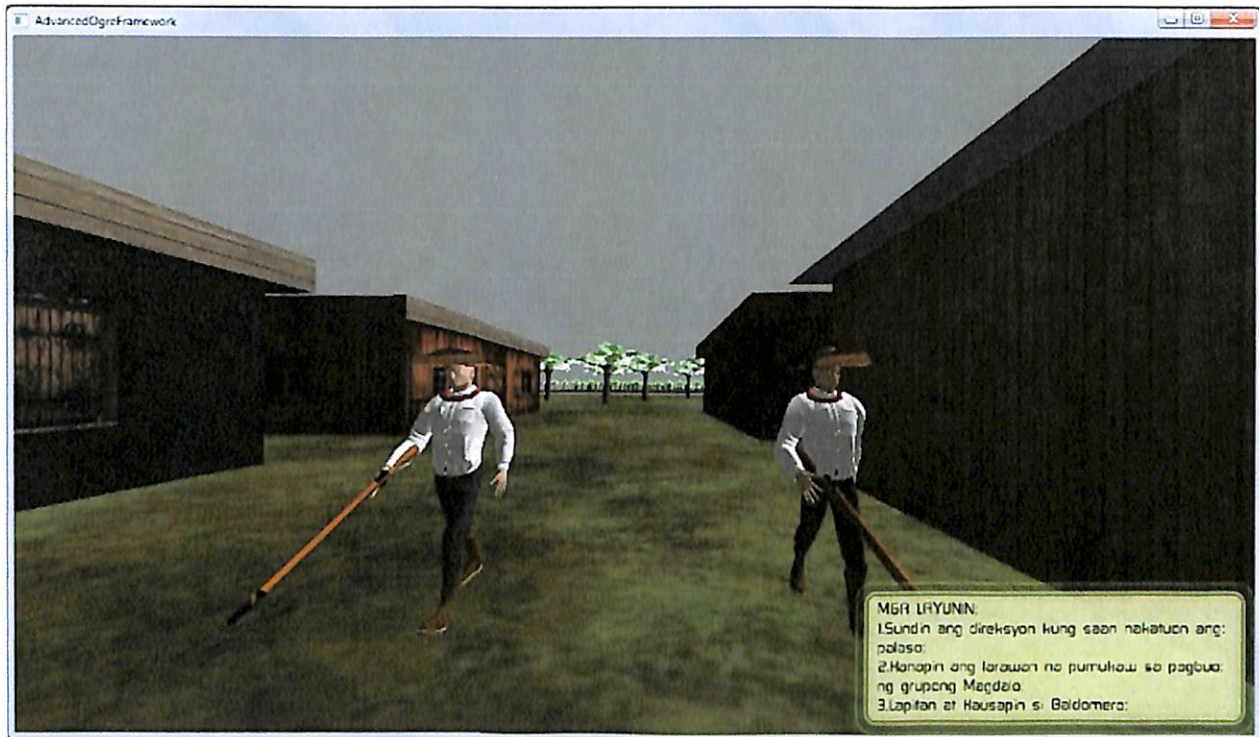


Figure 7. Sample characters

E.1. Character Modeling

The characters starts out as a single plane in 3DS Max. Creating the face and body of a character, the single plane created in the viewport will be extruded and looped using only the edges and vertices when the plane is going to be converted into a polygon. To convert the plane into a polygon, right click on the object and select the “convert to” context menu and select the “convert to polygon” button. When the object is converted into a polygon, the user will be able to select its vertices, edges and faces with options available to modify that selection. To be able to modify the selections, click on the object to modify and click the modify tab. Once in the modify tab, click on “Editable Poly” text found below the Modifier List. To create these edge loops, select an edge to loop and hold down the [SHIFT] button found in the keyboard and move the edge where you want it to be placed. This process is repeated as often to add faces into the geometry. As for combining multiple objects into one, select the object to combine; find the “Attach” button in the Edit Geometry rollout. After clicking the Attach button, left click on the object that is to be combined and its vertices and edges are going to be attached into the same model. To add an edge within a face, use the “Slice Plane” button found in the Edit Geometry rollout. When the character is almost done, the “MeshSmooth” modifier can help to smoothen out the model helping to avoid that edged look. Using loops, extrudes, scaling, rotating and moving the edges and vertices will help to make the models be defined and detailed to add into the overall look and feel of the game making it more distinguishable.

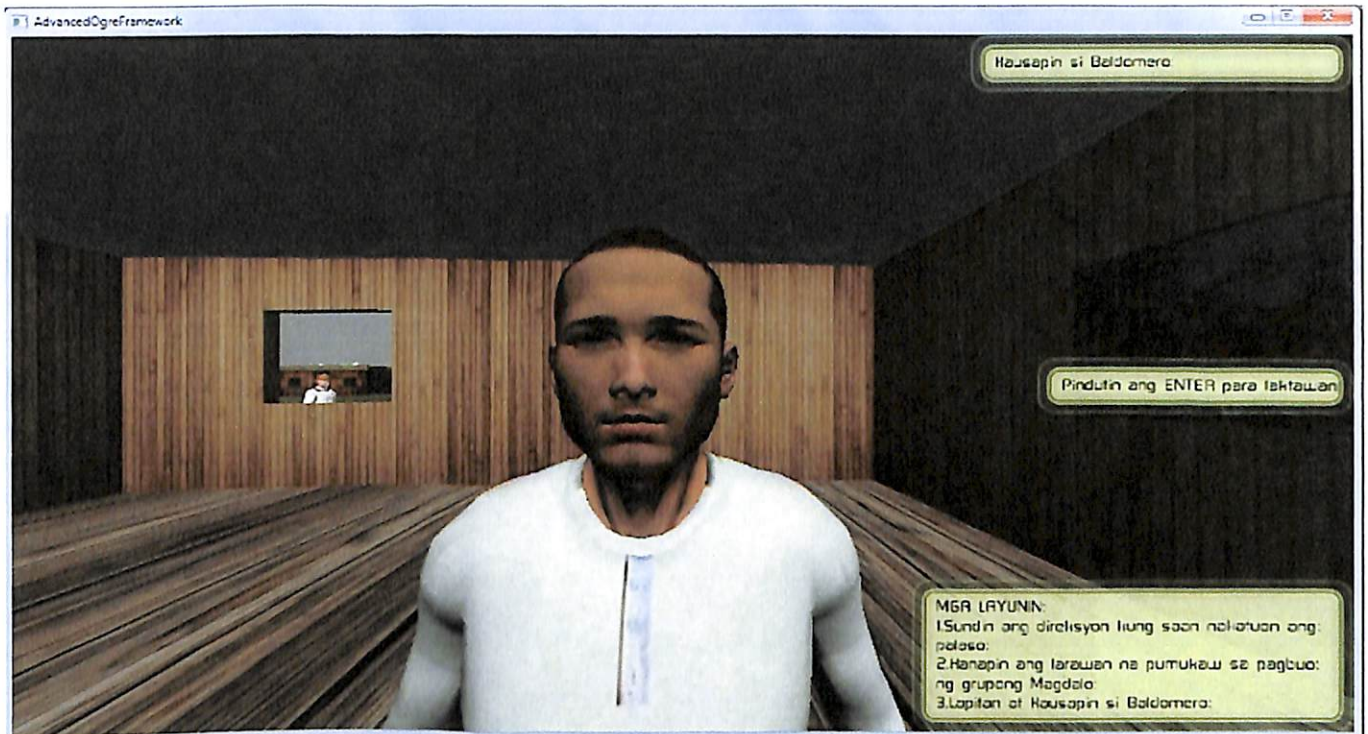


Figure 8. User is interacting with a character



Figure 9. Player is armed with a rifle

E.2. Character Texturing

3DS max has the ability to add textures to its models, and the frequently used are; diffuse, ambient, and displacement. There are also a number of ways to apply these textures. Designers have the option to apply UVW maps(under the Modifier List) to their models or they can use 3DS Max's built in Material Editor. On applying the maps on to models, select the model that is going to be applied with a texture, click [M] on the keyboard to bring up the "Material Editor". Once in the Material Editor, click on "Standard" under the Material/Map Browser and next click the Bitmap button. When the file dialog shows up, select the texture to be applied. Different textures are required for each face, for example, the eyes have a different texture to the humans skin texture, so a separate bitmap has to be applied on to that face. applying textures to a single face can be applied when the object is converted into a editable polygon.

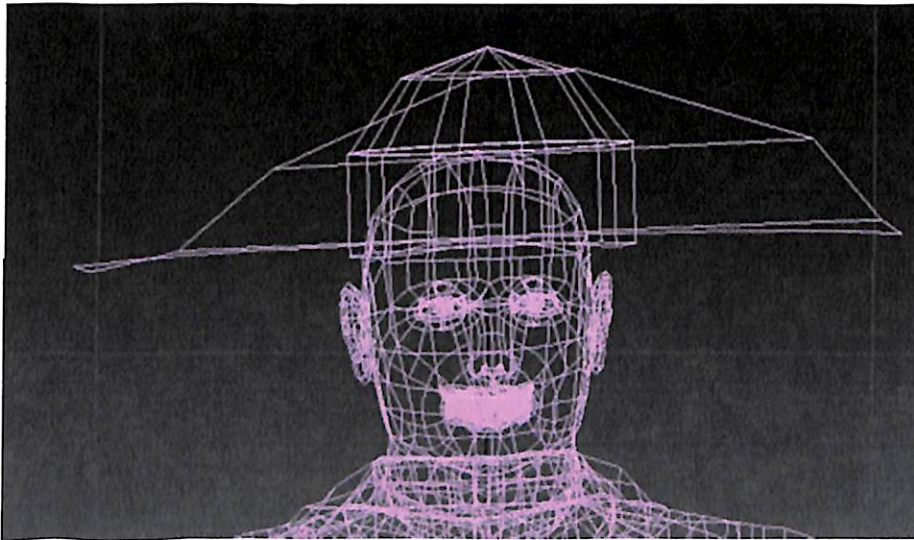


Figure 10. Character rendered in wireframe



Figure 11. Sample textured character

E.3. Biped Animation

On animating the Biped, the time slider becomes useful. Every move that is applied onto the biped will be automatically saved when the "Auto Key" button is pressed. Every biped's ligaments are moveable. By default, the time slider is set to "Frames per Second", and is set to 30fps, but this option can also be changed. To animate the walk, for this example, first; the left leg is moved forward on frame 0. Then on frame 5, the right leg is then moved forward while the left leg is now at the back and so on and so forth. This step is required to be refined to achieve the desired look of the animation.

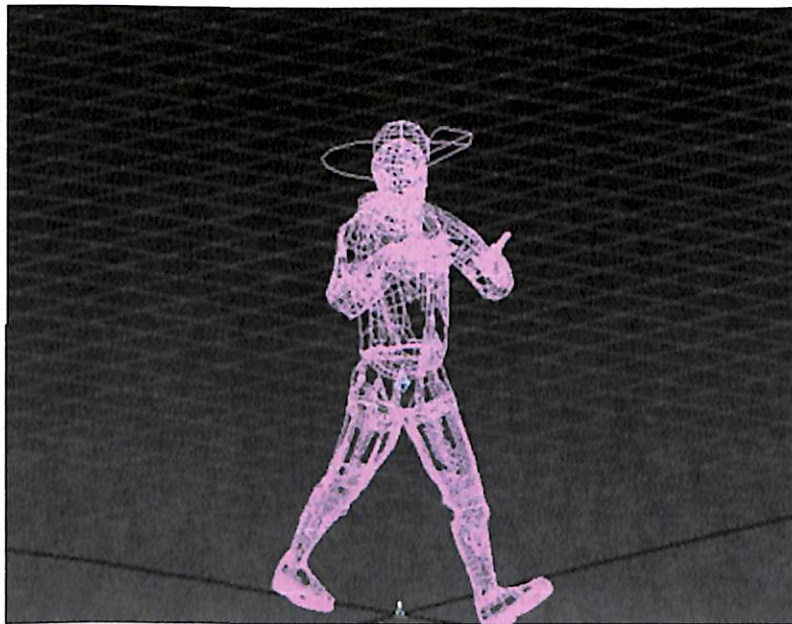


Figure 12. Character rigged with walking animation

E.4. Character-Biped Rigging

During the Biped Animation, the objects related to the character and its parts to be animated are made sure to be in a single model. When some of its parts are not combined, the parts are not going to be included with the model's Modifiers stack. When every character and its parts are in place and attached, the "Physique" modifier is going to be added. The Physique modifier helps to load the Biped into the object being animated. The Biped serves as the character's bones. The Physique modifier is found under the "Modifer List" within the Modify tab. Also, to animate the models the Biped object is needed. The Biped can be found under the "Create" tab, while under "Systems" button.

E.5. Mesh Animation Exporting

Under Mesh Settings, the biped object has to be selected. So when the meshes are exported, "character1.mesh" will have the "character1.skeleton" file which will be used to call these animations.

F. Populating the Game World

F.1. Terrain and Collision

The collision tools class is instantiated sets the player camera height. The player reacts if there are objects and meshes collide. The collision detection varies on the player height and the objects height if the object is lower to the height of the player it does not affect the player reaction and if the height of the object is higher than the height of the player then the collision detection is applied. The reaction on object collision is where the player step backs for every objects the player collides.

```
m_CollisionTools->calculateY(playerNode, false, true, SOLID_MASK);

    if(m_CollisionTools->collidesWithEntity(oldPos, playerNode->getPosition(), 2.5f, -
1.0f, SOLID_MASK))
    {
        playerNode->setPosition(oldPos);
    }
```

Scene management is where a particular scene is loaded in the game state. The loading part of the scene is under the function of `createscene()` where in a `dotsceneloader` class is used to load the exported scene.

```
DotSceneLoader* pDotSceneLoader = new DotSceneLoader();
    pDotSceneLoader->parseDotScene("main-terrain.scene", "General", m_pSceneMgr,
m_pSceneMgr->getRootSceneNode());
    delete pDotSceneLoader;
```

F.2. Loading - User controlled first person camera and other models

The `main-terrain.scene` is loaded and attach to the scene manager to be able to view the objects inside the scene. Inside the xml the position of the player is defined to set up the player camera position to the exact location.

```
<node name="info_player_start" id="462080736">
    <position x="-90" y="0" z="80" />
    <rotation qw="1" qx="0" qy="0" qz="0" />
    <scale x="1" y="1" z="1" />
</node>
```

The game camera is created in menu state, game state, and pause state. A default free look camera system is created if the camera is not attached to a node or entity. The camera system can be attached on the player for a first person camera view.

```
m_pCamera = m_pSceneMgr->createCamera("GameCamera");
    m_pCamera->setNearClipDistance(0.1f);
    m_pCamera->setFarClipDistance(300.0f);
```

The camera named GameCamera is created and attach to the scene manager. Also the camera can set the camera farness where in how many objects and entities he can load to an exact distance. The mouse dictates the x, y and z camera direction and different camera rotation.. The scene manager will use the entity from the ogre resources by creating the mesh while the scene node will be use to attach the entity that is loaded in the scene node to be able to view the mesh file that is created.

```
Quest1_Entity1 = m_pSceneMgr->createEntity("Human_1", "soldier1.mesh");  
  
Quest1_mNode1 = m_pSceneMgr->getRootSceneNode()-  
>createChildSceneNode("Human_1Node", Ogre::Vector3(41, 0, -5));  
Quest1_mNode1->attachObject(Quest1_Entity1);  
Quest1_mNode1->setScale(3, 3, 3);
```

Mesh Objects are loaded on the game through the Ogre::Entity and Ogre::SceneNode. The position of the entity can be set through Ogre::Vector3 to position exactly where the entity stands. The mesh can be also scaled on how small or big the entity is. The x, y, and z vectors must be provided to load the entity to a specific location.

G. Play Testing

The 1st year college students who are taking up history have tested the game. Most of the players who tested the game were comfortable with the controls and its gameplay. Some of the players who aren't even instructed yet of the game's basic controls, were able to control them easily because of their past experience of playing First Person Shooter type of games.

6. CONCLUSION AND RECOMMENDATION

6.1. Conclusion

Based from the total results gathered the players were able to follow the objectives of the game easily. The players are guided on what to do in the game's quests on each levels. They were able to relate the life of Emilio Aguinaldo to the story of the game. Through the cut scene videos they appreciated what is the main quest for different levels. They enjoyed the game as they were able to interact with objects and characters within the game provided with a specific audio for each characters. They find the game fun through game characters are animated which moves to different locations. One fourth out of the 30 players surveyed had a difficulty on the movement and position of the character. They were confused of the movement of both inputs for keyboard and mouse. Some of them were not totally gamer who knows the usual basic controls of first person games. They had a difficulty on identifying specific characters in connection to Emilio Aguinaldo. They suggested that the characters modeled must be in particular to the real faces and images of those characters. On the game graphics most of players were able to rate the game to a score of four. They liked the setting of the game where there are objects and entities within the game. Houses are textured to look like old.

In terms of the rating on the game's audio most of the players surveyed scored two. They cannot hear the audio of a specific character they interact on merely because the volume of the sound is not too loud. They suggested that there must be a speaker or an earphone to be able to hear clearly the audio

of the characters in the game. They also are able to identify easily the Katipuneros in the game, the ones who are modeled wearing white shirts, pants and red scarfs. Also a gun is attached to the arms of the soldiers. The game lights were okay for them as they can easily see and identify the objects and entities. The lights are positioned well within the game which makes it more clearly to see on their perspective. On the comments/suggestions area one surveyed player said to put more characters and interesting features to the game. The player wants to see many kinds of characters modeled and animated. Another player suggested, making the game more fun where to put different interactions and in game cut scenes. The player wanted to put a specific scene if there is interaction to the main character and to the entity interacted that most of the games today do. So that the exact story behind a specific character can be understandable.

Players were able to appreciate Emilio Aguinaldo's story and his contributions during the Philippine Revolution. While playing the game, players also did learn about the Katipunan's factions because of the game, from where did the Magdalo's name came from and who are those characters that Emilio Aguinaldo met during the revolution. In making the game more interactive, they've become well receptive of the game's story.

6.2. Recommendation

Pitch problem – feature disabled

The mouse' pitch within the game was disabled. It was disabled due to the implementation's persisting bug, wherein if the pitch is enabled, players wouldn't be able to hit the enemies, for example; in Quest 3: Battle of Zapote. The ray casts to a point in the scene are sometimes imprecise, when the player tries to fire the weapon it would sometimes hit the scene's terrain, later causing it to be destroyed. When the terrain is destroyed from the scene, this would also cause the collision library to fail and the player's camera node would fall below to its negative z axis.

AI

There are lots to explore when it comes to integrating AI libraries into Ogre. It is quite vast and requires much of an in-depth study to be able to integrate it into Ogre, and if possible implemented along with a physics engine. If the AI decides to move without any collision or physics engine, it will only decide to move along its z axis not minding the necessary yaw, pitch and roll. The AI needs to adhere to the game's physics; it is most important if the game's terrain has an uneven surface. Integrating Ogre with an AI library is broad in itself; the topic can be used in a different study wherein it deserves much attention and research.

Game's textures – RTT

Games created using Ogre can also utilize 3ds Max's Render-to-texture feature wherein it improves the game's overall look. With RTT, content designers could improve the game's shading, shadows, reflection and refraction. This in turn would lessen the usage of hardware resources required to render real time shading techniques.

Fix the bullet physics and ray casting for the war scene

Bullet physics comes to mind when developing first person shooter games. When bullets fire at a target, it should collide with an object and when obstacles are in the way, it should have been blocked by that obstacle. We also recommend to find a good bullet physics library to include in game development.

Model the characters in their physical likeness

The users testing the game suggested to model the characters with likeness to which the model is based on. With the right texture and an enough amount of polygons, the result could have been achieved.

Face talking animation

The proponents recommend the face animation for character talking. The modelled character opens and close its mouth. The audio of a specific character rely on the precision of the animation. Aside from the animation of the mouth is the animation on different skeletons on the character modelled face. The eyes and other movement of the face must be in connection.

Health System

Recommending on the health bar of the character. There is a life system on the main character and the enemies. Checking if the life is enough to battle with character enemies. There is a graphic user interface on the player's screen to check if its life is low or high.

Ammo System

The particular set of ammo or bullets per gun firing. There is a limitation on the gun's ammo where the player reloads after if the gun magazine is empty. Picking up of ammo's to incorporate it to the gun attached to the character and if the gun's ammo is empty the player can not fire bullets to the enemies.