Short Paper

"TAKIPSILIM": A Fantasy-Themed 2D Turn-Based RPG Developed Using Unity Engine

Paolo Angelo P. Capundan
College of Engineering and Information Sciences, Trinity University Asia, Philippines
paoloangelopcapundan_softeng@tua.edu.ph

Mark Steven P. Evangelista
College of Engineering and Information Sciences, Trinity University Asia, Philippines
markstevenpevangelista_softeng@tua.edu.ph

John Andrei K. Libutan
College of Engineering and Information Sciences, Trinity University Asia, Philippines
johnandreiklibutan_softeng@tua.edu.ph

Kurtz Warren G. Bariso
College of Engineering and Information Sciences, Trinity University Asia, Philippines
kurtzwarrengbariso_softeng@tua.edu.ph

Ferdinand R. Bunag
College of Engineering and Information Sciences, Trinity University Asia, Philippines
frbunag@tua.edu.ph
(corresponding author)

Abstract

Purpose – Role-Playing Games (RPG) are an iconic genre for both tabletop and video game mediums that remains relevant even to this day. With the popularity of video games, it was inevitable that RPGs would also appear in this medium. In this study, the researchers aim

to develop a Fantasy-Themed 2D Turn-Based RPG developed using the Unity Engine that will enrich and educate people about Filipino mythological creatures, folklore, cultures, and traditions.

Method - The agile methodology was used in game development, which provided the developer with the ability to deliver faster value, better output quality, and the flexibility to make the necessary game adjustments. The study was broken down into smaller modules and consistent idea collaboration was done together with the stakeholders. The development of TakipSilim was created using the C#-based Unity Engine Language, built on the Unity Engine. The game storyboard was used to familiarize the developers with the game's story progression and challenges.

Results - For the findings, the system's evaluation was conducted on twenty (20) gamers in terms of gameplay, efficiency, reliability, usability, compatibility, maintainability, and difficulty and was based on the ISO/IEC 25010:2011 standard. The game was also evaluated by IT experts using different criteria. As a result, a very satisfactory rating of 4.7 was given to readability, ease of program maintenance, low hardware resource consumption, and low complexity. The only Source Code Adaptation received a higher rating of 4.8, which was also very satisfactory.

Conclusion - The study reveals that the game developed was able to provide its users with the gaming experience to promote, enlighten, and educate people about the local culture through folktales, mythological creatures, and superstitions. The game has functions and features that have a unique storyline written by the proponents and proves that there is quite a demand for Filipino-themed games, though niche, in the market today.

Recommendations – The proponents recommend and encourage future researchers to take the initiative to expand the horizons in game development revolving around Filipino culture due to the demand for the game revolving around the mythos.

Research Implications - The study has shown that the agile software methodology provided the developer with the ability to deliver faster value and better output quality. Also, the system evaluation was implemented based on the basic ISO standards which results in a very satisfactory rate. Lastly, creating RPG games from the ground is very challenging for amateurs, but they it able to grow and hone their skills as researchers and game developers.

Keywords – Role-Playing Games (RPG), Filipino-themed games, Unity Engine, Agile Methodology

INTRODUCTION

With the popularity of video games, it was inevitable that RPGs would also appear in this medium. An example of a very successful video game RPG series is the Ultima Series which came out in 1981, and have adapted the mechanics of Dungeons and Dragons (DND) which became the staple for how future video RPGs are made in western countries. In the east, specifically in Japan, the series Final Fantasy was starting to get traction in 1987. Compared to their western counterpart, Japanese Role-Playing Games (JRPG) are more linear and story-driven with an emphasis on character development while the western RPGs were mostly focused on character customization and placed the player in the shoes of their Self-Insert in the fictional world (Pelletier-Gagnon, 2018).

The influences of RPGs in the Philippines can be seen in both tabletop games and video games. An example of a Filipino-made RPG that gained popularity was the RPG game, Mamayani. An RPG that focused on educating the players about the achievements of powerful women in history, and thus it showed that women of the time played a crucial role in liberating the Philippines from the Spanish Regime as much as the men did. The game was developed using RPG Maker, a popular and easy-to-use engine for creating in the style of JRPG. Though there were limitations because of the engine used, the developers were still able to create a game with a unique personality and style while taking inspiration from popular RPGs like Earthbound (Cerda, 2021; Pelletier-Gagnon, 2018). The RPG tabletop scene on the other hand has Tadhana, which was an RPG inspired by Filipino mythological creatures, folk tales, and urban legends and it was made by a single developer, Nathan Briones. Its gameplay was heavily influenced by DND, but the creatures and lore of the game are inspired by our local mythology. Using the same principle in classic tabletop RPGs with Job Classes, and going on Campaigns, however, the gameplay used a card mechanic instead of the classic dice roll mechanic in tabletop games to make it easier for new players to adapt to the genre. The project was developed using Unity software to promote, enrich and educate people about the local culture through folktales, mythological creatures, and superstitions. Specifically, the study aims to produce an interesting and compelling game with a fictional story about local mythological creatures, folktales, and superstitions by following the formula of storytelling according to the Hero's Journey (Campbell, 1990).

The developed RPG will also be subjected to evaluation based on ISO -25010:2010 criteria. The video game also used the integration of RPG's core rules, mechanics, elements, and features. It was focused on creating interesting, entertaining gameplay and story for the player to interact with and experience within the developed video game. However, the developed game will be strictly single-player and will not include an option for multiplayer in any way. The game will be developed strictly for Personal Computers (PC) as the researchers believe that it is not feasible considering the time and financial constraints to create a mobile or console version of the game. With the richness of several local

mythological creatures, superstitions, and folktales, the researchers picked a selection of interesting entities. Finally, the minimum game length for completion will be at least two hours of gameplay.

In this regard, the specific problem were; how to develop a mobile RPG game application that highlighted fictional local mythological creatures, folktales, and superstitions, and to design a game that provided a stylized approach on how to depict mythological creatures, folktales through art and animation, promote, enrich and educate people on Filipino mythology.

LITERATURE REVIEW

The Implication of an Interesting Story for RPG Game Development

In the book "The Hero with a Thousand Faces" (Campbell, 1949) his research into the common patterns that he found in stories that centered around stories with a hero embarking on a quest, and it is now iconically coined as "The Hero's Journey". The Hero's Journey depicts the progression and development of the Protagonist in the story and how they tackle the obstacles they will encounter in their journey. A few examples of these stories that used the same formula would be The Wizard of Oz, The Hobbit, Star Wars, and The Lion King. The Hero's Journey (Campbell, 1990) was divided into an RPG with incredible game design and mechanics will be interesting for players, however, having a lackluster story with no consistent flow will overshadow the judgment of players and they will reach a point where they would lose interest in the game before the mechanics have time to shine. It is for this reason that an interesting story which pulls in players right from the beginning was crucial in developing an RPG. This is the main concept that the game followed. The "The Hero Journey" has guided the developer on how the game will be played and accomplished.
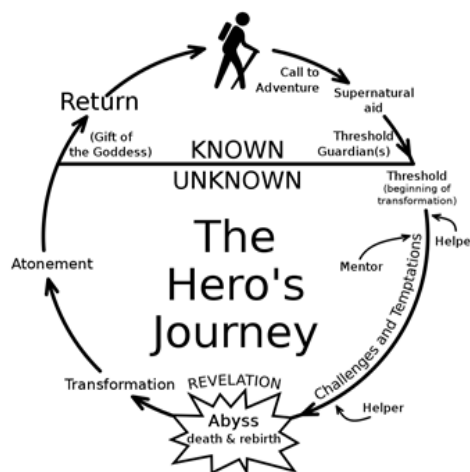

Figure 1. The Hero's Journey

A classic RPG requires the player to go on an adventure as the main character. For an interesting game to go on or have high ratings, a game will need a good matching storyline. Most story-driven games usually follow the formula of The Hero's Journey by (Campbell, 1990) (See Figure 1) where, as the game progresses, the game's narrative also develops. The main purpose of having a story in the game is to help the player feel more involved and immersed in the game. Giving the player a way to connect with the virtual world, the adventure, and the characters of the game.

Modern RPG Implementations

Modern RPG developers still use the same formula when designing their games, even though the concept was made in 1974 it was still the bread and butter of RPGs. Even the most niche RPGs would use an Experience Point (EXP) system and Health Point (HP) system while having their unique twist to them, a good example of this would be the Indie game Undertale which used the EXP system and Leveling system as a story motivator instead of a way to simply show character stat growth. The Turn-Based mechanic as well was improved throughout the years, where at the beginning it only dictated which character or enemy acts next, the Final Fantasy series made their Turn-Based mechanic act in real-time. RPGs evolved on a larger scale when the Internet was introduced and the creation of Massively Multiplayer Online RPG (MMORPG) like World of Warcraft had players from all over the world, to play simultaneously on a server going on quests together. To make the game more interesting, the developer utilized the EXP to quantify the game avatar's life experience and progression during the game. After accomplishing a mission, surviving an opponent's attack, and overcoming given obstacles, additional experience points will be given. For HP, the game was also designed to accumulate the character's amount of hit, damage, and stamina loss during a mission.

Current State of Game Development (Triple-A and Indie Development)

When we talked about the Game Development industry, two things came into mind and those were the Triple-A industry and the Indie (Independent) Development. We often hear stories of developers and artists working in the AAA industry who were overworked, underpaid, and given bad working conditions (Brogan, 2022). This was very evident in the most anticipated release of 2020 which was Cyberpunk 2077. CD Projekt Red announced their new intellectual property Cyberpunk 2077 last 2012, and the game started development from 2016 until September 2020 (Zarzycka, 2022). In the initial development of the game, CD Projekt Red promised that they will not submit their staff to "the crunch", however during the 4 years of development the game underwent the staff was put into several crunch periods where they were overworked without any additional pay just to meet the deadlines the company has given them (Cote & Harris, 2021). A common criticism

and stark contrast with Indie developers is that AAA developers like Electronic Arts (EA) rely heavily on Quantity over Quality. This is the act of developing games that are copy-paste of previous games with minimal features added which is evident in their sports games (NBA, FIFA, and Madden series) and which they still charge full price for their games (Thompson, 2012).

Indie Game Developers are starting to receive the recognition and global success they deserve for their games over the last few years. A few noteworthy names would be Toby Fox the sole developer of Undertale, Eric Barone AKA Concerned Ape the sole developer of Stardew Valley, and the recent recipient of the Best Independent Game Awards Supergiant games which developed Hades has made a name for themselves in the Indie Development Scene (Fiadotau, 2018). These indie games that are developed by single people or small teams can beat the sales against AAA developers that are backed by multi-million-dollar companies. The developers are inspired by the ideas of Indie Development, and a team with very limited financial resources and technical support, was able to finish the project.

Key Factors in Game Development

Video games have been around in the future, the development of video games was evolving into a more complicated product with greater graphics, better controls, more advanced hardware, and many more. The video game industry, especially its software segment, was one of the more rapidly growing industries in the economy (Zackariasson and Wilson, 2012). The creation of video games follows five key factors used in developing video games and these were games engines, programming languages, graphic design software, audio engines, and artificial intelligence. Game engines are the frameworks for AAA developers and Indie developers to build and create video games. In a sense, a game engine provided the tools necessary for rendering graphics, collision detection, sound management, scripting animation, lighting effects, and many more proving very useful to the developer. Examples were Unity which the developers were using for this game development, Unreal Game Engine, Game Maker: Studio, and many more.

Programming languages were used to write the codes of the game which lets the developer communicate with the computer hardware and develop the game's logic and algorithm. Used in the game development is the A-star Algorithm which is commonly used in identifying the path that provides the least cost from the initial node to the goal node. The available programming language that was suited for game development was C++, C#, Lua, Java, Python, Objective-C, and Action Script. Effective game development will eventually boil down to the skill of the programmers behind them. The graphic design software was typically used for creating UI, Character Designs, Game Environment Designs, and many more from animations to images, for the result of making the typical user gaze

in awe in front of the screen. As many users evaluate that graphics were key to having an enjoyable experience when playing video games.

Graphic design software does its job by giving the right number of tools to create artwork, sprites, splash art, and concept art for developers. Examples of software that were used today were Adobe Photoshop, Pixelmator, GIMP, Inkscape, Marmoset Hexels, and many more. Audio engines, Sounds were also important as the logic of the code and the aesthetic of the graphics for making video games. Auditory stimulation was used to give the player feel more and build immersion right in front of their face. Some gaming companies hire orchestras to produce soundtracks, as well as hire popular voice actors which gave the game personality and life. Popular audio engines for making games were FMOD, Wwise, Miles Sound System, Fabric, Astound Sound, and many more. Lastly, Artificial Intelligence in video games has existed throughout the years of gaming. As it helps keep track of the flags and events for the player while playing, it will decide many factors of the game in making progression, results, enabling events, guiding NPC and so much more. In a sense, Artificial Intelligence was used for responsive and adaptive behavior for the developed game (Videgaín & Sánchez, 2021).

METHODOLOGY

During the design phase, the developers utilized the Agile Model (Figure 2) approach for incorporating all concepts, modules, and designs into the game's development. Using this model, the developer was able to adapt and respond to the game adjustment the user requested.
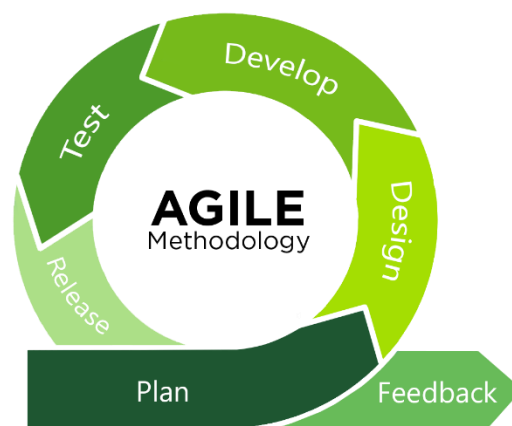

Figure 2. Agile Methodology

The developer used Agile Development Methodology; it is currently one of the most popular software development methodologies. The program was broken into tiny incremental builds and delivered in iterations with Agile development. For the preparation of working functionality with each build, all activities were separated into short periods.

Because each stage was checked and thoroughly reviewed before moving on to the next stage, issues were detected and fixed quickly. Agile Methodology provided the developer to deliver faster value, with better output quality and flexibility to make the necessary game adjustment. This resulted in efficient workloads that focused and prioritized the important part of the game's concepts, modules, or designs. It trimmed down the entire development time and delivered a high-quality output. To coherently understand the development process, a storyboard, mechanics review, and concept design were created.
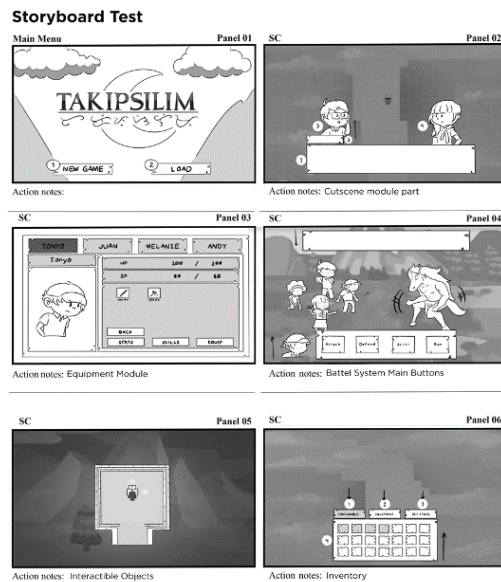

Figure 3. Game Storyboard

Game storyboard (shown in Figure 3) was used to familiarize the developers with the game's story progression and challenges. After conceptualizing the project with the mechanics, assets, and progression needed, the development phase will begin. The game was developed using Object-Oriented programming on C#. A program of this scale requires multiple scripts running simultaneously while also restricting other scripts from running unless called. Without Object-Oriented programming, it wouldn't be possible to construct a coherent game with an understandable script. Coroutines were used because the script needed to wait for the player to select an input.

Unity Engine provided several unique C# based functions which were used to manipulate Game Objects in the game to function according to the scripts that interact with them. Scripts inside the game were all instanced based, where the activation of scripts follows restrictions before launching. Through instanced-based scripts, scripts that utilize the functions of other classes can access them across other Game objects and Scenes. Github was used to streamline the file transfer process when collaborating with fellow developers in the project. This provided an efficient way of reverting a current build or updating to a newer build with easy-to-use tools. The mechanics were implemented with

C# scripts, while the creation of graphical assets was done in various software applications like Adobe Photoshop, Adobe Illustrator, Adobe Animate, Clip Studio Paint, and Aseprite.

The Overworld module was the hub for most of the modules presented in the system. Depending on the player input, they were able to have multiple access to the main menu module and its sub-modules as well as the Player Movement module with the Interact able modules. To determine the error in the system, the developer conducted testing on each module and presented the system to show if it meets the objective and is capable of displaying correct data to the users. Other module functioned well during testing like login, saving, game control, updating, and removing. The design test in every browser and the size of the screen during testing worked properly. The developer detected errors on some major modules such as in the Overworld module, coding, rendering, and uploading. The developer also encountered an error in selecting game levels. The selected level also overlapped with the other levels. Another bug in the system was the delayed execution of the game control. After detecting errors, these were then appropriately fixed. The system was evaluated by gamers and IT experts. Gamers used the ISO-25010:2010 evaluation criteria, while IT Experts in the field of game development checked the game performance and quality assurance.

Operation

On the first level of the Overworld module, the player can move or access the Main Menu, or they can proceed to explore the world using the Player Movement module. The Main Menu module has four submodules, the status menu module, inventory menu module, notes module, and save module. The Status Menu Module and Inventory Module both have complex program algorithms because of how intricate their inner working process is; the programming methods used in these modules required in-depth knowledge of core programming. The Main Menu module has four submodules, the status menu module, inventory menu module, notes module, and save module.

The Status Menu Module and Inventory Module both have dedicated system process flow because of how intricate their inner workings are. The Main Menu Module can be closed by pressing the ESCAPE button on the player's keyboard. The Player Module has access to multiple modules that it can interact with, namely the Walkable and Impassable module for movement around the map, the Interactable Module, and the Random Encounters Module that was directly linked to the Battle Module. The Status Menu Module contains the Stats of the four playable characters in the game, which were tied to Scriptable Objects for each character. The Status Menu has several sub-modules, the Stats Module, which was the default state where it just shows the numerical representation of the character's stats, and the Equipment Module, which was the most complex of the three sub-modules for it altered the character's stats directly by changing

equipment of the selected characters, and the Skills Module which shows the currently learned skills of the character selected.

The player swapped characters with a button press, and the screen changed accordingly depending on the active character selected. The Equipment Module enabled the player to equip, change equipment, or swap equipment around with characters. There was only one piece of equipment that has special parameters and restrictions, and those were the weapons. A character was only able to equip one type of weapon, and if the player selects a weapon that does not match the character, they will receive a message that says that the weapon was invalid. For the other equipment types, the only restriction was that if the current equipment was equipped on a different character, they will be asked if they wish to swap the equipment between two characters. An equipped piece of equipment will have an indicator of who it was currently equipped on. Once a piece of equipment has been placed on a character, it will alter their stats according to the equipment's stats, and it will change accordingly if the equipment has been changed. The Inventory Module was the container for any items the player picks up or buys with their in-game currency. Loot can be in the form of three types, Consumables, Equipment, and Key Items.

Consumables were items that were removed from the inventory upon use, while equipment was permanently stored in the inventory until sold or dropped; lastly, Key Items were story-relevant items that cannot be removed from the inventory. Out of the three sub-modules, the Consumables module was the most complicated since it deals with altering Character Scriptable objects. The Consumables Module does not only store the consumable items the player had picked up, but it was also where the player may use said items. When the player decides to use an item, they are prompted to select a character to use it on. If the character selected already has full HP, the player is shown a message that the character has full HP and the item will not be consumed, and they are allowed to select a different character to use the item on. If the player can be healed, the item will then be consumed, and the selected character will recover X amount of stats depending on if the item is an HP or SP recovery item. There was a special parameter where if the character was instead overhealed from the item, they will only be able to heal up to MAX HP or SP, and the overhealed value will be discarded.

There was also a special consumable item "Revive items" which can only be used if the character has 0 HP left and their isAlive parameter was false, their character will recover an X amount of HP and the KO status will be removed. The Interactable Objects Module all used the same type of GameObject in the game, where it utilizes Trigger Colliders when a player overlaps their collider with the Trigger Collider, thus initializing the module. Most of the Interactable Objects activate when the E key is pressed on the keyboard; however, some of them activate by simply walking over the interactable area. There are three sub-modules where the trigger does not require a key press, and those

were the Warp Trigger Module and the Cutscene Trigger Module. The Warp Trigger Module acts as a teleporter to redirect the Player Avatar to different locations on the map it was connected to. The Cutscene Trigger Module was used to activate the Cutscenes that were on certain parts of the game to continue the story of the game, the Cutscenes themselves require the player to press Z to continue the cutscene, and the module requires continuous key presses until the cutscene was completed. These two modules also activate the Level Loader Module, which activates the Fade-In and Fade-Out animation to act as transitions between scenes. Finally, the Random Enemy Encounter is a collider on all walkable paths on the overworld, where each tile is considered a collidable object and has a fixed percentage of an enemy encounter when collided with. If the RNG trigger hits a certain number based on the probability of the encounter in that area, a battle sequence will activate.
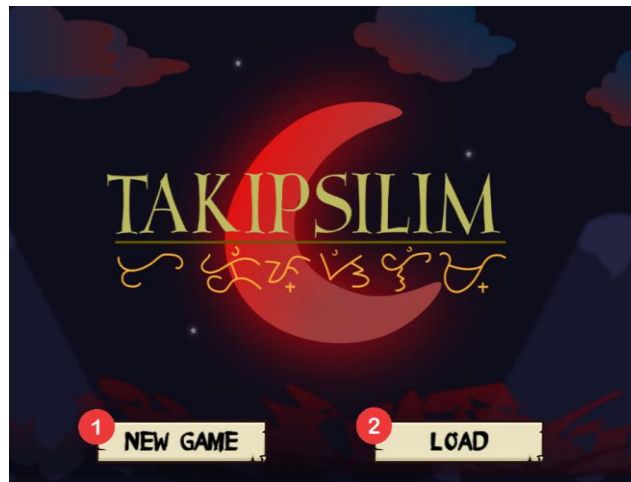

Figure 4. Shown in the figure is the game Title Screen

The Title Screen shown above has two modules that can be operated using the Left-Mouse Button. (1) The New Game Button, which was used to start the game; if there was an existing save found, the player will be reminded if they were sure if they want to start the game from the beginning. (2) The Load Game Button; if there was an existing save file on the disk, the game will be loaded on the last saved file; however, if there was no save file found, the player will be told that there was no existing saved file and will not be able to proceed.

Figure 5. Map Select Screen shows different game levels

Map Select (Figure 5) it's where the player selected what area or chapter they wanted to play in. All modules on this screen were interactable using the Left-Mouse Button. Most of the Chapters were locked in the beginning, but as the player progresses through the game, the chapters were unlocked accordingly. (1) The Save Button where the player saved their current game progress when they wish to stop playing and continue at another time; it is interactable using the Left-Mouse Button. This will create a file on the disk that will store all the data and progress of the player. (2) Load button; it functions similarly to the Load Button in the Title Screen. (3) The Map Select panel; where the player selected the chapter/map they will be teleported into.

Figure 6. Battle Mode

The Battle System handles all fight scenes inside the game that follows the turned-based RPG rules. The controls were simple, as the player only needs to use the left mouse button and think about their next move or predict enemy attacks. Most interactable objects were buttons that have designated corresponding functions. The Battle System is

designed to spawn the Playable Characters on the left (4) while the enemies will spawn on the right (1), as shown in Figure6.

RESULTS

The system's evaluation was conducted on twenty (20) gamers in terms of Game-play, Efficiency, Reliability, Usability, Compatibility, Maintainability, and Difficulty. This is based on the ISO/IEC 25010:2011 standard. The table shows these criteria and the evaluation result (Table 1).

Table 1. Gamer's Level of Acceptance of the System

Characteristics	Mean (M)	Verbal Interpretation
Game-play	4.8	Very Satisfactory
Performance Efficiency	4.8	Very Satisfactory
Reliability	4.7	Very Satisfactory
Usability	4.8	Very Satisfactory
Compatibility	4.7	Very Satisfactory
Difficulty	4.8	Very Satisfactory
Maintainability	4.8	Very Satisfactory
Overall Weighted Mean	**4.7**	**Very Satisfactory**

Table 1 shows the summary of the results of the evaluation made by twenty gamers. Having a very satisfactory rating with A WEIGHTED MEAN OF 4.7 are Game-play, Performance Efficiency, Usability, Difficulty, and Maintainability. The Reliability and Compatibility received a rating of 4.7.

Table 2. IT Expert's Level of Acceptance of the System

Characteristics	Mean (M)	Verbal Interpretation
Readability	4.7	Very Satisfactory
Ease of Program Maintenance	4.7	Very Satisfactory
Adaptation of the Source Code	4.8	Very Satisfactory
Low Hardware Resources Consumption	4.7	Very Satisfactory
Low complexity	4.7	Very Satisfactory
Overall Weighted Mean	**4.7**	**Very Satisfactory**

The game was also evaluated by IT experts using different criteria. A very satisfactory rating of 4.7 was given to Readability, Ease of Program Maintenance, Low Hardware Resources Consumption, and Low Complexity (Table 2). Only Adaptation of the Source Code received a higher rating of 4.8 which was also very satisfactory. Both ranked

ratings showed that developed games were strongly acceptable by the game evaluators. The evaluation result was very important because it showed how satisfied the gamers were in playing the games and this affected the reputation of the game and its developers. These high ratings demonstrated player enjoyment, this is considered the most important attribute of any game.

DISCUSSION

The main objective of this study was to develop an RPG game that introduced Filipino fantasy-themed; mythological creatures, folklore, cultures, and traditions to make gaming interesting and informative. The game was developed using the concept of RPG. The game size was small, but it utilized the latest genres and subgenres currently popular in media. These were determined by an extensive gathering of online information on the type of game to be designed. The developer was able to create the game and determine its performance, evaluation by gamers and IT experts were conducted. The gamers used the ISO-25010:2010 criteria and gave a very satisfactory rating of 4.7 which indicated that the game was highly accepted by the gamers. The criteria were based on ISO-25010:2010. The game was also evaluated by IT experts, and the result was a very satisfactory rating of 4.7, which shows that the game composition was developed properly. The game was responsive when used on other devices and can be used on an android mobile platform with an app icon on the mobile screen. It provided a good navigational experience to the user on the different game adventures available. The assets and code used in the development of the game were all made from scratch and the design of the main characters is interesting and appealing including the nor-playable characters in the game. The map design fits well with the theme of folklore and Filipino culture, and the placement of the mythological creatures was based on their characteristics and habitats.

CONCLUSIONS AND RECOMMENDATIONS

In the light of the findings of the results, it was concluded that the game developed was able to provide its users the gaming experience to promote, enlighten, and educate people about our local culture through folktales, mythological creatures, and superstitions. It was also shown how fascinating the fictional plot, characters in the story, and the enemy designs were based on, though stylized but accurate, in depicting the mythological creatures in terms of teaching players about Filipino Culture. The game is functional and features a unique storyline written by the proponents and proved that there was quite a demand for Filipino-themed games, though niche, in the market today.

DECLARATIONS

Conflict of Interest

All authors declared that they have no conflict of interest.

Informed Consent

All authors declare that this study is covered by the exceptions in the need of informed consent as no personal details were obtained.

Ethics Approval

The research was cleared by the Trinity University of Asia – Institutional Ethics Review Committee, a Philippine Health Research Ethics Board with approved protocol code 2021-1st-CEIS-Capundan-V2.

REFERENCES

Brogan, R. (2022). The Digital Sweatshop: Why Heightened Labor Protections Must be Implemented Before Crunch Causes the Backbone of the Video Game Industry to Collapse. *Texas Review of Entertainment & Sports Law, 23*, 13–55.

Campbell J. (1949). *The Hero with A Thousand Faces*. New World Library.

Campbell, J. (1990). *The Hero's Journey. Joseph Campbell on His Life and Work*. Harper & Row.

Cerda, C.M.C (2021). Towards a "Filipino" Video Game: Teaching Filipino Culture and Identity for Video Game Development. *Transaction of the Digital Games Research Association, 5*(3), 43-64. https://doi.org/10.26503/todigra.v5i3.122

Cote, A. C. & Harris, B. C. (2021). The cruel optimism of "good crunch": How game industry discourses perpetuate unsustainable labor practices. *New Media & Society*. Advanced online publication. https://doi.org/10.1177/14614448211014213

Fiadotau, M. (2018). Indie Game. In N. Lee (Ed.), *Encyclopedia of Computer Graphics and Games* (pp. 1-4). https//doi.org/10.1007/978-3-319-08234-9_188-1

Hamby, Z., & Hamby, R. (2019). *The Hero's Guidebook: Creating Your Own Hero's journey*. Creative English Teacher Press.

Pelletier-Gagnon, J. (2018). 'Very much like any other Japanese RPG you've ever played': Using undirected topic modelling to examine the evolution of JRPGs' presence in anglophone web publications. *Journal of Gaming & Virtual Worlds, 10*(2), 135–148. https://doi.org/10.1386/jgvw.10.2.135_1

Thompson, H. (2012). HEY RUBE: Blood Sport, the Bush Doctrine, and the Downward Spiral of Dumbness. Modern History from the Sports Desk (Book). (2004). *Publishers Weekly, 251*(28), 58.

Videgaín, S. & Sánchez, P. G. (2021). Performance Study of Minimax and Reinforcement Learning Agents Playing the Turn-based Game Iwoki. *Applied Artificial Intelligence, 35*(10), 717–744. https://doi.org/10.1080/08839514.2021.1934265

Zacchariason, P. & Wilson, T.L. (2012). *The Video Game Industry: Formation, Present State, and Future*. Routledge.

Zarzycka, A. (2022). Cyberpunk 2077 and transgressive failure. *The Journal of Gaming & Virtual Worlds, 14*(1), 27–37. https://doi.org/10.1386/jgvw_00049_1