COMMANDER: A WEB-BASED GAME OF THE GENERALS-INSPIRED GAME WITH A.I

A Thesis

Presented to the Faculty of the
College of Science
Technological University of the Philippines
Ayala Blvd., Manila

By

JERALD C. CASIMSIMAN
MHARI ALLEN E. LIMIN
GABRIEL C. MARAON
CHRISTIAN NOEL C. NACARIO

In Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Science in Computer Science

June 2024

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines | Tel No. +632-5301-3001 local 608 Fax No. +632-8521-4063 | Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	TUPM-F-COS-16-TAU
		Revision No.	00
	THESIS APPROVAL SHEET FOR THE UNDERGRADUATE PROGRAMS OF THE COS	Date	07012022
		Page	1 / 1

APPROVAL SHEET

This thesis hereto entitled:

COMMANDER: A WEB-BASED GAME OF THE GENERALS-INSPIRED GAME WITH A.I

prepared and submitted by **JERALD C. CASIMSIMAN, MHARI ALLEN E. LIMIN, GABRIEL C. MARAON, CHRISTIAN NOEL C. NACARIO** in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** has been examined and is recommended for approval and acceptance.

PROF. EDWARD N. CRUZ
Adviser

Approved by the Committee on Oral Examination with a grade of **PASSED** on JUNE 7, 2024.

PROF. ARIEL TOMAGAN
Member

PROF. PERAGRINO AMADOR JR.
Member

PROF. DOLORES L. MONTESINES
Department Head/Chair

Accepted in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.**

Date: June 13, 2024

DR. JOSHUA T. SORIANO
Acting Dean

ACKNOWLEDGEMENT

The researchers wish to express their sincere gratitude to all individuals that facilitated the initiation and contributed significant efforts to the success of this study. Foremost, deepest thanks are extended to the Technological University of the Philippines-Manila for providing an exceptional educational opportunity and to God Almighty for endowing wisdom to everyone and fostering the pursuit of knowledge.

A heartfelt gratitude is given to the research advisor, Professor Edward Cruz, whose invaluable advice and steadfast support were pivotal in the research's completion and success. The mentorship provided by Professor Cruz has been both an honor and a privilege.

Sincere appreciation is also extended to the College of Science faculty, particularly to Professor Dolores L. Montesines, for her comprehensive guidance throughout the Bachelor's Program. Gratitude is also further expressed to the panelists for the generous contribution of their time and knowledge.

Furthermore, the researchers convey heartfelt thanks to friends and family for their unwavering support and assistance throughout the study.

The resolute understanding, patience, and confidence in the researchers' abilities were nothing but crucial to the study's success. The numerous hours of help, insightful discussions, and sustained motivation provided along the journey are deeply appreciated.

ABSTRACT

This study titled "Commander: A Web-Based Game of the Generals-Inspired Game with A.I." examined the development and performance of an AI-enhanced version of the traditional Filipino board game, Game of the Generals. The purpose of this study is to remind fellow countrymen of the impact and competitiveness of local board games especially when adapted into a modern medium which is the web. The game was designed and created with custom map terrain and made use of gun sprites for the playable units as a more engaging way to quickly remember the hierarchy. The software, adhering to ISO 25010 standards, incorporated unique features such as customizable map and piece designs. Rigorous testing verified that the application met design specifications and operated smoothly across various devices. User feedback yielded an overall rating of 3.37, classified as "Very Acceptable," with high marks for functionality, performance efficiency, and portability. Key features included strategically hidden enemy pieces, diverse map layouts, and options for classic or modern sprite models. Developed using the Godot Game Engine, Visual Studio, Adobe Premiere Pro, and Adobe Photoshop, the game demonstrated high quality and provided an engaging, strategic gaming experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figures

LIST OF APPENDICES

CHAPTER 1

THE PROBLEM AND ITS BACKGROUND

In this chapter, the project's overview, problem statement, and significance will be discussed. General and specific objectives, as well as the scope and limitations are also outlined. This section will provide a summary of the research, its societal and personal implications, and a brief description of the themes explored in the study.

Introduction

Games have been proven to be effective teaching tools for education. Even Plato suggested using games to teach kids in one of his books namely 'The Republic' (Maroney, 2001). Games such as board games, have long been a staple in Filipino households, providing multi-generational entertainment and educational benefits. Board games help kids develop their social and cognitive abilities while instilling in them patience, strategy, and turn-taking (Ehrenfeld, 2022). Meanwhile, for adolescents to adults, board games are mostly used as a common feature of family get-togethers and social gatherings for the sole purpose of relaxation and social interaction (Stemmle, 2023).

Popular board games such as the 'Game of the Generals' (GG) is a popular board game originated in the Philippines is also known as the "Filipino Chess" due to its emphasis on strategy and tactics just like chess. Both players will have an army of their own just like in chess, but two armies engage in a military simulation, and participants strategically maneuver their pieces, symbolizing various military ranks and units, in an attempt to outsmart their opponent and seize the enemy's flag. The twist lies in the fact that the movements of these pieces are concealed, keeping their ranks undisclosed to the opponent. This introduces an element of deduction and bluffing, enhancing the complexity of the gameplay (Pasola, 1973).

With the rise of modernity and online transition of most things involving entertainment, even the GG boardgame was adopted into either a personalized software edition or through an online medium where people can play it with their various devices and online platforms. An example of this is the mobile game "Game of The Generals Mobile" released on 19-Nov-2020 by Mawkins Entertainment available on Google Play. (Mawkins Entertainment, 2020). However, only a few adaptations were made online that are still playable and active, which became the inspiration for this study.

In contrast to the existing game mechanics and overall game experience GG could offer, this study would introduce new game features that will distinguish it from the known popular games. Features include a new set of units as well as their respective hierarchy, a larger playing area with random terrain gimmicks and map effects, unit power-ups, etc. This study would make use of reinforced learning algorithm for its Computer Bot opponent, an AI algorithm that emphasizes ways of improving performance through trial-and-error experience. While there have been advancements in creating artificial learning systems, the majority of these systems rely on a knowledgeable "teacher" to guide them on how to react to specific training stimuli. However, such systems face limitations, especially when obtaining the necessary training examples is expensive or unfeasible. Reinforcement learning, on the other hand, enables autonomous systems to learn from their own experiences rather than solely relying on guidance from knowledgeable teachers.

Background of the Study

Game of the Generals (GoG) is a Filipino-made board game that uses realistic ranks in the military as pawns for war in order to practice and sharpen the players' strategical and tactical skills. It has the moniker of "Salpakan" in Filipino which comes from the word "Salpak" or "to clash". It is designed and loosely based around chess where the players not only battle using tactics and proper piece placement, but also by strategically creating tension between the two players due to the unknown ranks of each piece on the board, thus creating a sort of psychological warfare between the two all, while being judged by a third party which is the arbiter (Paciente, 2023). The game itself simulates the feeling of war and being in the "fog of war" because a player can see and place their pieces where they see fit, they can only see where the opponent places their pieces but doesn't know which piece is which.

In order to create a similar but also novel feel to the Commander board game, we must first understand the rules of the board game that it is based off. First setting up the table, the players can place their pieces on any space on their side of the board as long it is within the first 3 rows, then the players will choose who goes first. When moving, a piece can only forwards, backwards, and left and right in one space, and when a piece is put onto a space where an enemy piece is sitting in, a battle will occur in which the arbiter checks which of the pieces rank higher in order to determine who wins, if they are in the same rank, they will both be eaten. As for the rankings in order of highest power: Five star general, Four star general, Three star general, Two star general,

One star general, Colonel, Lt. Colonel, Major, Captain, 1st Lieutenant, 2nd Lieutenant, Seargent, and Private. There are also special pieces such as the spy which can beat any piece except for the private, and the Flag that can be capture by any piece and can only capture another player's flag (Mortensen, 2017).

Artificial Intelligence can be defined as the ability of a machine to have an information processing system capable of generating new useful information, and to create new algorithms in order to solve specific tasks (Suleimenov et al., 2020). In the case of the study, it will be used as the decision maker of the opponent, in order to enhance the experiences of the player in specific areas such as strategic planning and tactical placement by limit testing your own skills. By taking in various factors at play such as what pieces the player has left, what terrain the pieces could be on, or other external factors, the AI can prepare an optimal move for most of the possible scenarios that can play out in the board game.

Objectives of the Study

The main objective of this project is to develop a web-based multiplayer game inspired from the Game of the Generals and implement an A.I. opponent in case the user cannot find another player to play with.

Specifically, it aims:

- To design and create a game inspired by the existing ruleset of Game of the Generals but with an added twist to differentiate it from Game of the Generals.
- To develop an A.I. adversary that can realistically simulate an actual human opponent with multiple levels of difficulty to help teach new players and provide a fun challenge for veteran players.
- Evaluate the functionality, usability and design of this study using ISO 25010.

Significance of the Study

The significance of the study lies in its multifaceted contributions to the world of gaming and beyond. Firstly, by developing "Commander," a web-based multiplayer game inspired by "Game of the Generals" with an integrated A.I. opponent, the project addresses the modern need for accessible, engaging, and socially connected gaming experiences. This is especially crucial in

the context of today's technologically interconnected world, where online gaming has become a prevalent form of entertainment and social interaction.

Scope of the Study

The scope of the study involves the technical and creative aspects of game development, including programming, graphic design, and user interface design. Additionally, it delves into the realm of artificial intelligence, focusing on creating a responsive and adaptable opponent that can enhance the game's overall appeal. Throughout the project, careful attention will be given in ensuring that the A.I. opponent offers both learning opportunities for newcomers and a formidable challenge for experienced players. The study will contribute to the gaming community by offering a novel web-based multiplayer experience inspired by a classic game, "Game of the Generals," and by providing a unique A.I.-driven gaming experience for users seeking solo play or competitive practice.

Delimitations of the Study

There are a few limitations to our study. Firstly, this program shall be a web application instead of a standalone executable in order to allow more people on various platforms to be able to play the game.

CHAPTER 2

CONCEPTUAL FRAMEWORK

This chapter presented the review of related literature, related studies, conceptual model of the study, and the operational definition of terms.

Review of Related Literature

Artificial Intelligence in Games

Artificial Intelligence (AI) has been a popular topic in the gaming industry for years. AI is used in games to create non-player characters (NPCs) that can interact with players and provide a more immersive experience. In recent years, AI has been used to create more realistic and challenging NPCs that can adapt to the player's actions. "Research on the Application of Artificial Intelligence in Games" discusses how AI is used to create NPCs that have a certain degree of self-learning and self-optimization ability to improve the playability of the game (Zhang et al., 2022). "Artificial Intelligence in MOBA Games: A Multivocal Literature Mapping" presents a multivocal literature mapping of available research that focuses strictly on the use of artificial intelligence approaches in Multiplayer Online Battle Arena (MOBA) games, one of the most popular esports genres and the one most widely used for game AI and game analytics research (Costa et al., 2023). "Research on Artificial Intelligence Algorithm and Its Application in Games" describes how game artificial intelligence (AI) has become the technical core of improving the playability of a game and the main selling point of game promotion, deepening the game experience realm (Tang et al., 2020). Finally, "Measuring Intelligence through Games" summarizes that games of various kinds are commonly used as benchmarks for “narrow” AI research, as they are considered to have many important properties. We argue that many of these properties carry over to the testing of general intelligence as well. We then sketch how such testing could practically be carried out. The central part of this sketch is an extension of universal intelligence to deal with finite time, and the use of sampling of the space of games expressed in a suitably biased game description language (Schaul et al., 2011).

In board games, Artificial Intelligence (AI) has been used to create more challenging and realistic gameplay. "The effectiveness of intervention with board games: a systematic review” conducted a systematic review of 71 studies to examine the effectiveness of board games and

programs that use board games. The study found that board games can improve the understanding of knowledge, enhance interpersonal interactions among participants, and increase the motivation of participants (Noda et al., 2019). "The application of artificial intelligence in board games" [2] discusses how AI helps people play against computer opponents in board games (Jiang, 2023). "A Data Driven Review of Board Game Design and Interactions of Their Mechanics" presents a data-driven review of board game design and interactions of their mechanics (Samarasinghe et al., 2021). These articles provide a comprehensive overview of the application of AI in games and its impact on the gaming industry.

Educational and Cognitive Aspects

Board games are games that are usually sought after for their entertainment value as well as the fact that two or more people can play at a time either together or against one another. In the Philippines in specific, there has been an increase in demand for board games in the past years as shown by the growth in shipments as well as the country reportedly relying more on imports in order to meet the demand of the Board Games Market (6Wresearch, 2023).

Aside from being a good source of entertainment, they can also be used as a medium for learning. Effective games allow players to organize information in a way that makes it concrete, as well as providing avenues to link previously gotten information to new information and combine them with one another to come to a conclusion. In the case of team-based board games, qualities such as critical thinking and empathy were also a requirement and resulted in and improvement in those same qualities as well as their capacity for collaboration and creating working relationships (Treher, 2011).

In a study where the researchers were tasked to create different games with the goal of being educational in 3 different aspects: Knowledge revision, development of strategic thinking, historical empathy. The first game “Mystery to History” focused on knowledge revision and determining which school the information given to the player relates to, to which the player is then given an overview of as well as the option to organize their collected information. The second game “Identity” which had historical empathy as the goal, required the players to play the roles of high school students in different situations where they had to cooperate with one another. The third game “Nastavlenije” had the goal of understanding what it took in order for the school systems in Serbia to develop properly as well as exploring the difficulty of school organization and

management, since the players were made to manage limited resources and satisfy a number of conditions in order to win. And the last game "Danilo's game" is a role play card game that makes use of a GM (game master) as well as a player that takes the role of an ordinary man in Serbia in the end of the 19th and early 20th century where they experience different situations and make decisions in order to affect the course of their life in the game (Rajković et al., 2017).

Game of the Generals Influence

"Games of the Generals" is a strategy board game that has had a significant influence on the Filipino culture. The game has inspired the formation of more than 2,500 GG Clubs 1. It is considered an educational war game that teaches board game strategy and tactics to Filipino locals back in the day (Studocu, 2021). The public debut of The Game of the Generals was on February 28, 1973. After the game was released marked the first influence in many Filipinos, as Filipino chess players were outraged, believing that Pasolas (the inventor) was attempting to belittle or replace chess. (Studocu, 2022). This is due to the fact that in its three years, three million Filipinos tried the game, was introduced to 33 different countries, and was the first and last board game to over outsold chess at local bookstores and supermarkets. It is said that the GG had a hard time carving out a place in the world or thinking sport as it had a history of continued cold war between the local chess organizations. There was even an incident where the Philippine Chess Federation threaten to ban its players from joining a GG-Chess tournament since various chess grandmasters showed interest in playing the game after Pasola tried to promote his game in big-prize laden chess tourneys. This particular board game gave Filipinos a feeling of historical pride up until this day as it achieved something that no other local board games could ever replicate (The Times Journal, 2015).

The game was so popular and meaningful in the country that a certain General Foundation specifically the General Miguel Malvar Foundation commissioned the development of an online adaptation of this popular Filipino board game in order to remember Miguel's legacy since Miguel Malvar was a Filipino commander who fought in both the Philippine Revolution (Spanish colonization) and the Philippine-American War (Malvar.net, 2023).

Ranking System

Since the game simulates armies at war trying to outflank and outmaneuver each other, the game consists of varying units with corresponding ranks and functions. The player consists of a set of 21 pieces where a piece with a higher rank can defeat any piece with a lower rank, except for the spy, which can defeat any piece except the private. The pieces are angled to conceal their rank or insignia from the opponent (Rubis, 2011). All of this will also be inherited by Commander but with a different take on the ranking system, terrain and number of units.

The Private

In the U.S. Army, the rank of private is typically the starting point for soldiers entering basic combat training without a four-year college degree. The term "private," when associated with military officer ranks, was first documented in the mid-1700s. It originates from the Latin word "prīvātus," which means "private" in the sense of being owned by an individual. Soldiers can advance from this entry-level rank to private second class, then to private first class, and subsequently to specialist, before achieving the rank of corporal (Oliver et al., 1983).

Figure 1
Private Rank Symbol

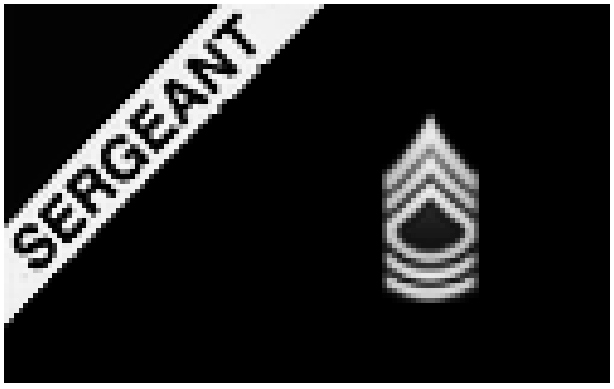
In the game, Games of the Generals, the Private is also considered the lowest rank in the hierarchy and can only eliminate the flag. But it also has a special function where it is the only unit in the game that can win against the spy (Mortensen, 2017).

The Sergeant

In the U.S./Philippine Army, a sergeant is described as "a noncommissioned officer ranking above a corporal." Sergeants are responsible for supervising soldiers' daily duties and

typically lead a team or section slightly larger than that of a corporal, usually consisting of up to four soldiers. Due to the diverse responsibilities and tasks associated with the rank, several grades of sergeant have been established. Historically, the Air Force had six sergeant grades, whereas the Army and Marines each had five (NHHC, 2014).

Figure 2
Sergeant Rank Symbol

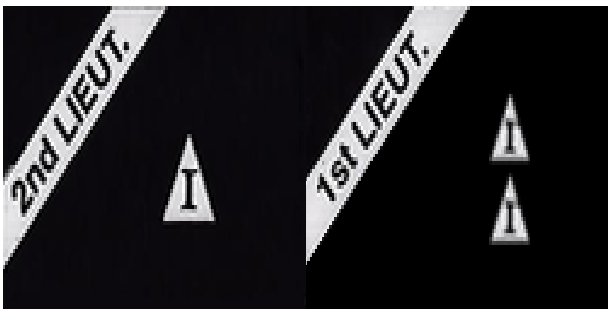


In the game, Games of the Generals, the Sergeant is considered 2nd to the lowest rank in the hierarchy and can only eliminate both the flag and a private ranked unit. (Mortensen, 2017).

The First & Second Lieutenant

Said militaries recognizes two lieutenant ranks: second lieutenant and first lieutenant. A second lieutenant is considered "the lowest commissioned officer rank" in the Army, Air Force, or Marine Corps, while a first lieutenant is positioned above a second lieutenant but below a captain. The lieutenant rank is the first officer level that can be achieved. Second lieutenants typically lead platoons consisting of 16 to 44 soldiers. In contrast, first lieutenants command more specialized platoons and companies, which are small units made up of a headquarters and two or three platoons. The term "lieutenant" dates back to Middle English, first appearing between 1325 and 1375 (Dictionary, 2010).

Figure 3
1st & 2nd Lt. Rank Symbol

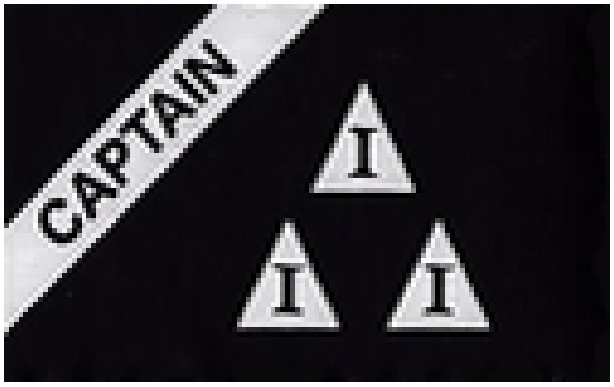


In the game, Games of the Generals, the First & 2nd Lieutenants are ranks considered be just above the lowest two in the hierarchy and can eliminate whichever lower ranked unit based on the hierarchy (such as private, sergeant, 2nd Lieutenant. (Mortensen, 2017). The 2nd Lieutenant's equivalent in Commander is the SMG while Assault Rifle for the 1st Lieutenant.

The Captain

Captains rank above lieutenants in the military hierarchy. Defined as "an officer ranking above a first lieutenant and below a major" in most armies, the term "captain" has its origins between 1325 and 1375. It is derived from the Late Latin word "capitāneus," meaning "chief" (Dictionary, 2010).

Figure 4
Captain Rank Symbol



In the game, Games of the Generals, the Captain is a rank considered to be just on the middle of the hierarchy and can eliminate whichever lower ranked unit based on the hierarchy (such as private, sergeant, 2nd/1st Lieutenant) (Mortensen, 2017). The Captain's equivalent in Commander is the Shotgun.

The Major

A major holds a commissioned military officer position ranking below a lieutenant colonel and above a captain. They commonly lead brigades and task forces. The term "major" was first observed between 1350 and 1400 (Runser, 2021).

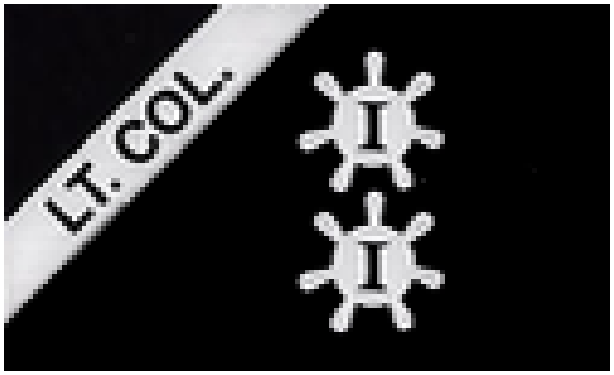
Figure 5
Major Rank Symbol

In the game, Games of the Generals, the Major is a rank considered to be just on the middle of the hierarchy as well and can eliminate whichever lower ranked unit based on the hierarchy (such as sergeant, 2nd/1st Lieutenant, Captain) (Mortensen, 2017). The Captain's equivalent in Commander is the Machine Gun.

The Lt.Colonel

Lieutenant Colonels primarily function as commanders of battalion-sized elements, typically comprising around 800 soldiers. However, they can also fulfill roles as staff officers at the brigade level or higher, assisting in unit planning, resourcing, and operations. In their capacity as battalion commanders, Lieutenant Colonels hold ultimate authority over all activities within the battalion. They are accountable for all actions and decisions made by the battalion, bearing responsibility for both its successes and failures (Nagele 2023).

Figure 6
Lt. Colonel Rank Symbol



In the game, Games of the Generals, the Major is a rank considered to be one of the highest before the star ranks (Generals) and can eliminate whichever lower ranked unit based on the hierarchy (such as 2nd/1st Lieutenant, Captain, Major) (Mortensen, 2017). The Lt. Colonel's equivalent in Commander is the Rocket Launcher.

Multiplayer Gaming Communities

Videogames cater to a wide audience of players. Certain games lend themselves well to the creation of large communities that provide support and camraderie to other fellow players of the same game. These communities form due to their passion for the game and let fellow players interact and organize with each other (MetaEngine, 2023). Videogames in general skyrocketed even more when the world was sent into lockdown due to the covid pandemic. As many as 79% percent of people that participated in a survey were playing videogames with the total time spent playing increasing by 26% (The NPD Group, 2020). An article in the BBC mentioned that "The explosive growth of gaming during the pandemic has shown that many have found a new outlet for much-need connection in isolation" (Chang, 2020).

A survey of select World of Warcraft (WoW) players in Australia concluded that a majority of their participants drew social support from their relationships with other players (O'Connor, Longman, White, & Obst, 2015). However, in another study by a different set of researchers, they found that specifically to browser-based games, a high importance of community doesn't automatically lead to a low importance of solo-playing as the two factors mirror the different needs of the individual players as it is up to the players to decide whether certain aspects of a particular video game would be given a higher or lower prioritization (Schultheiss, Bowman, & Schumann, 2008).

Web-Based Game Development

Web-based game development is a rapidly growing field that has gained significant attention in recent years, reflecting the growing demand for accessible and interactive gaming experiences on the web platform. A literature review "A guideline for game development-based learning" provides a guideline for game development-based learning (GDBL) methods using game development frameworks (GDFs). The study identifies relevant features of GDFs and presents

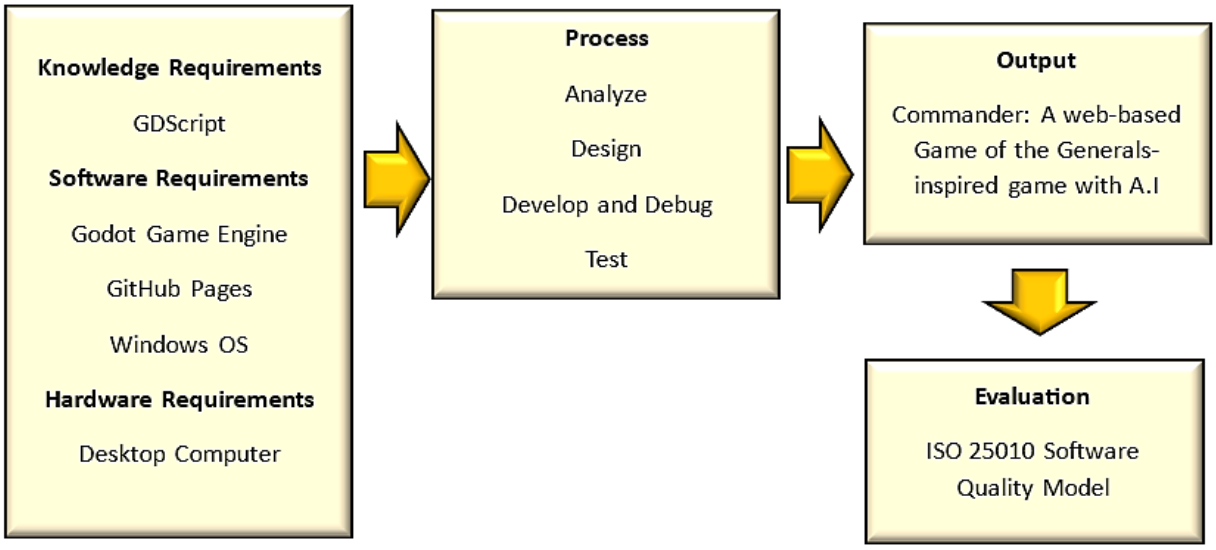
empirical evidence on the educational effectiveness of the GDBL method (Wu & Wang, 2012). In the study "The Future of Web and Mobile Game Development", it said that with the public availability of HTML5, numerous developers have been exploring its potential for web development. The purpose of this study is to provide an overview of the implications for the game development community. The paper assesses emerging HTML5 elements and features in JavaScript. Specifically, emphasis is placed on WebGL, Canvas, and WebSockets, empowering developers to showcase their creativity through image manipulation, the creation of 3D environments, and the facilitation of real-time interaction (Curran & George, 2012).

Research conducted by (Ma & Wang, 2017) has investigated the complexities and prospects of developing games for the web, underscoring the necessity to enhance performance and ensure compatibility across different browsers. (Haro et al., 2019) examined the incorporation of WebSocket's to facilitate real-time interactions in online multiplayer games, noting its significance in boosting player engagement and communication. Additionally, (Ansong & Ong, 2020) have discussed how HTML5, and JavaScript are instrumental in achieving cross-platform compatibility, enabling the creation of games that operate smoothly on a variety of devices. These studies collectively advance the comprehension of the technical intricacies, obstacles, and breakthroughs in web-based game development, offering crucial perspectives for both developers and scholars.

Game engines are also essential in game development, offering a suite of tools for crafting and deploying interactive experiences. Various studies have examined their impact and evolution. (Kelleher & Pausch, 2005) explored the educational applications of the Alice game engine, while (Chen & Pan, 2013) assessed the integration of game engines in programming education. (Anderson et al., 2018) investigated game engines' role in creating VR experiences. (Müller & Wufka, 2018) provided insights into game engines' ongoing development and their implications for the gaming industry. These studies collectively advance the comprehension of the technical intricacies, obstacles, and breakthroughs in web-based game development, offering crucial perspectives for both developers and scholars.

Conceptual Model

Figure 7
Framework Concept



The figure shows the Conceptual Framework of the Commander: a web-based game application. This has been based on the input, process, and output. This framework will serve as a guide information for the reader to understand the concept of the study. In order to create a successful website, the requirements needed in the study are knowledge, hardware, and programming software. As shown above in figure 1, the development of this website requires knowledge of GDScript as the main programming language. The software requirements include Godot as our game engine for our game, we will then host it using GitHub pages as a website and we will use the windows as our main operating system. The hardware requirements encompass a desktop computer. The development process involves analyzing requirements, designing the system, developing, and debugging the code, and performing testing. The output is Commander: A web-based Game of the Generals-inspired game with A.I that can be played by player vs player and player vs computer. The system's quality can be evaluated using the ISO 25010 Software Quality Model, which assesses characteristics like functionality, reliability, performance efficiency, usability, and maintainability.

Definition of Terms

Arbiter - A person that has the authority to decide disputes. In the project it is the third party that decides which pieces come out of each battle alive.

Artificial Intelligence - The ability of a machine to mimic and perform cognitive functions similar to human minds. In the project it is the backbone which decides the actions of the enemy computer.

Board - A place to put your physical pieces on usually made of wood or plastic. In the project it is the space in which the terrain of the application is generated and is the basis how units can move.

Game of the Generals - An educational war game board game made in the Philippines by Sofronio H. Pasola Jr. In the project it is the template on which the other features of the commander board game are based off of.

GDScript - custom scripting language that is used in the Godot computer game engine. In the project it is the programming language used by the researchers to create the applications features.

Godot - Allows users to create 3D or 2D games. In the project it is the framework used for creating the application.

ISO 25010 - A quality model used in a product quality evaluation system. In the project it is the guidelines used by the researchers in order to check the application's quality.

Pieces - Item of a particular type, usually forming one of a set. In the project it is the units that the player controls in order to beat the enemy.

User Interface - It is the point where human-computer interaction occurs. In the project it is the allows the player to interact with the application.

Web based platform - A program that is mainly accessed over a network using HTTP. In the project it is the where the commander application will be put to play on.

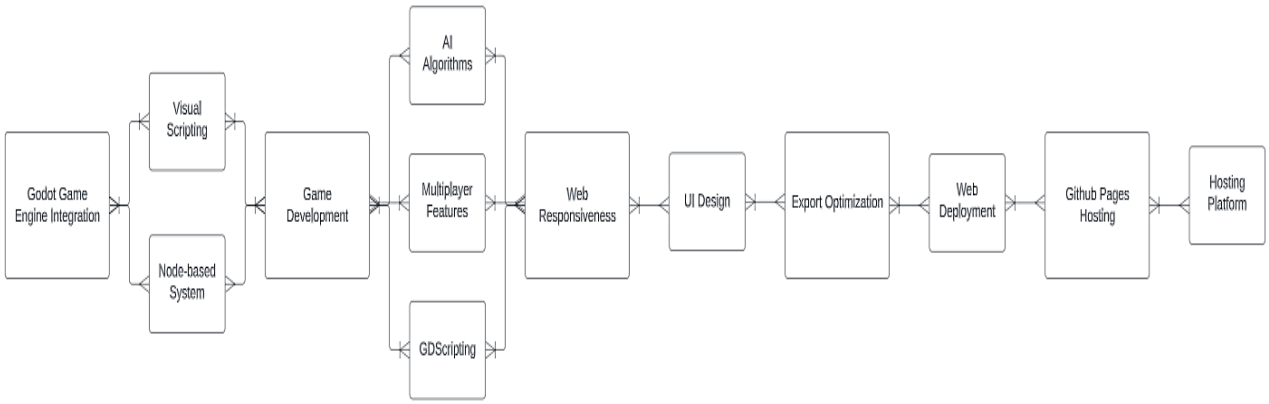
CHAPTER 3

METHODOLOGY

This chapter provides a detailed discussion on the project design, development, operating and testing, and evaluation process of the system methods as well as the procedures that will be conducted by the researchers, as well as the proposed flow of the steps to be completed in order to accumulate information essential for the study.

Project Design

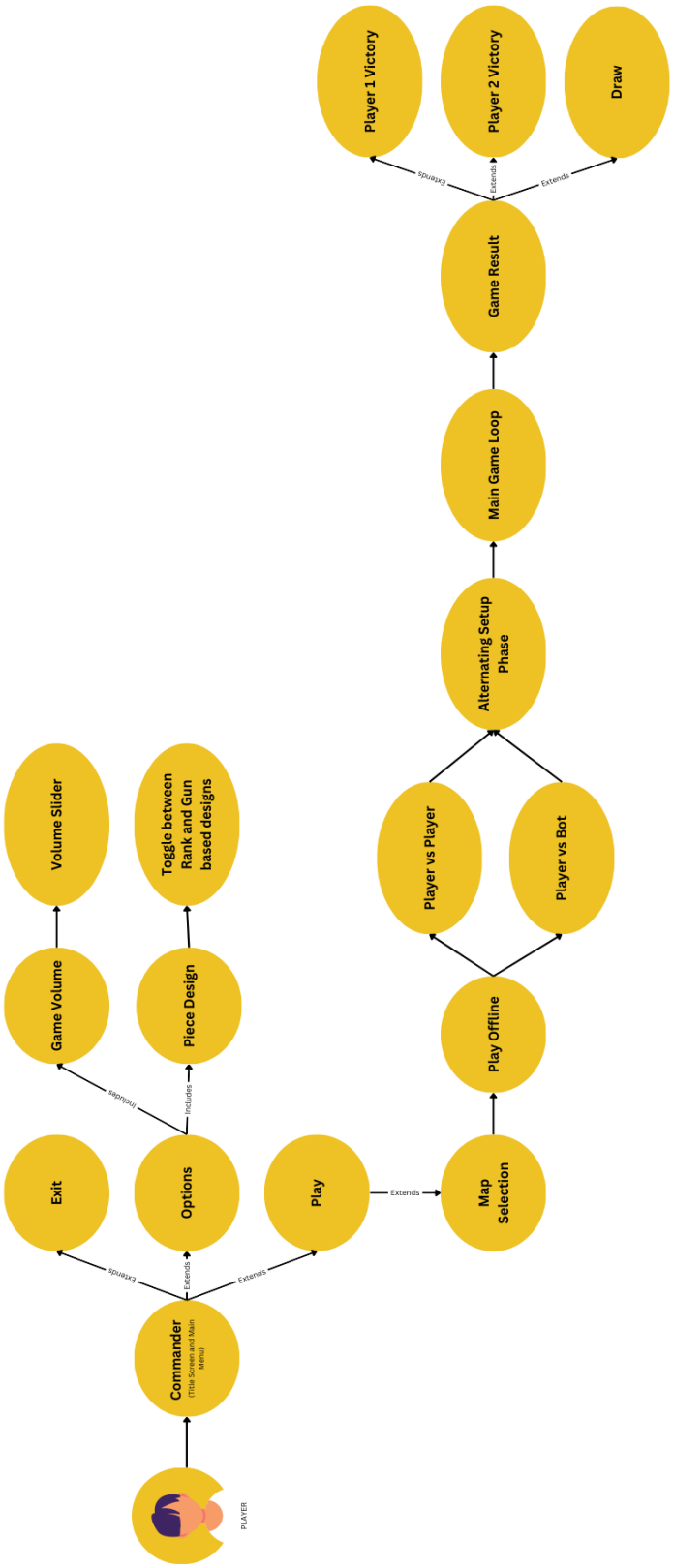
Figure 8
Block Diagram of Commander Web-based Application



This research for "Commander: A web-based Game of the Generals-inspired game with A.I" involves utilizing Godot as the game engine and hosting the game on GitHub Pages. The design will employ Godot's visual scripting and node-based system to create an architecture that seamlessly integrates the intricacies of the gameplay and the AI-driven opponents. The user interface will be designed with web responsiveness in mind, ensuring an immersive gaming experience. The game development will leverage Godot's GDScript language to implement AI algorithms, multiplayer features, and export capabilities optimized for web deployment. GitHub Pages will be utilized as the hosting platform, providing a straightforward and accessible way for players to engage with the game directly through their web browsers. The design aims to merge the strategic depth of the game with the capabilities of Godot and the convenience of GitHub Pages for a compelling and easily accessible web-based gaming experience.

Software Design

Figure 9
Use Case Diagram



One of the project designs for Commander used the Unified Modeling Language (UML), specifically a Use Case Diagram. The diagram served as a summary of the user's details and how their interaction with Commander went.

Figure 9 shows the flow of the game from the player's perspective. The game begins at the Title Screen, which also serves as the Main Menu of the game. The Main menu contains buttons for Play, Options, and Exit. Clicking the Play button allows the player to then select a map to play on. After that they are given a choice whether they want to play the map with another player as PvP, or with a bot as PvE.

After the game ends, the players are then brought to a Results screen showing either the player win, lose or had a draw with the AI. They can then press the Menu button to go back to the Title Screen / Main menu in order to play another game.

The settings button allows the player to adjust the Music Volume as well as which piece design they would like to use; Rank Based (Which is derived from the original Game of the Generals piece designs) or Gun Based which is the design implementation of the researchers and the default setting for the game. And lastly the Exit button allows the player to quit the game.

Table 1

Use Case Titles and Description of the Index of Commander

Use Case No.	Use Case Title	Use Case Description
UC01	Title Screen and Main Menu	This case begins when the application is launched. It shows the Title screen of the game as well where the user can access all of the other cases and commands in the game. It includes the Map Select, Settings, and Exit.
UC02	Map Select	This case is where the player can choose the map to play on.
UC03	PvP or PvE Selection	This case is where the user can choose to either battle against another player or a trained AI in a local game.

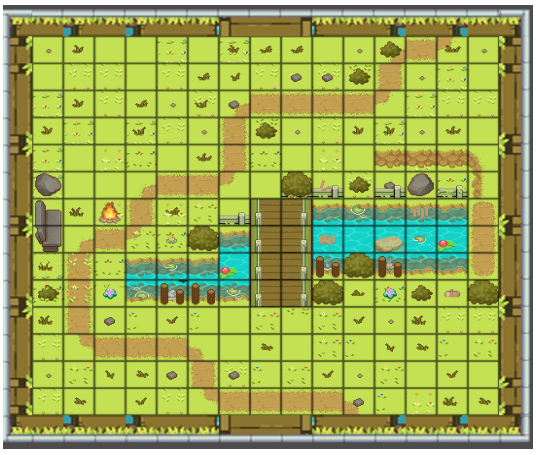
UC04	Main Gameplay Loop	This case simulates the gameplay of the game itself.
UC05	Options	This case allows the user to configure their Game Volume. level and the Piece Design used.
UC06	Exit	This case terminates the application

As for the gameplay of Commander itself, it is rather simple. The gameplay begins with a Setup phase where alternating players place their respective pieces on their side of the board which are then hidden to the opponent depending on whose turn it is. After managing to defeat the other player's HQ they are then brought to the Victory screen, or the Draw screen if both HQ's were eliminated at the same time.

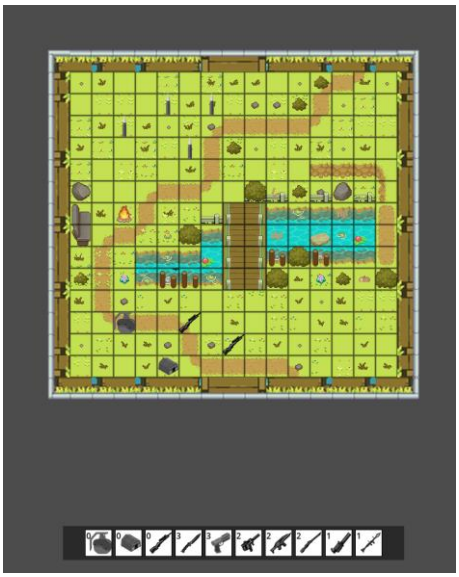
Story Board

The flow of the gameplay in the Commander web application when navigating through the options are shown in Figure 5. From the start the user is given 4 options with the information on the game mechanics being the first for rule familiarization purposes. The next option being the main game itself in finding a match either through other live players searching online, or by facing against an AI for faster match times. The rest can be seen as choices the player can make after finishing a match between either of the previous options.

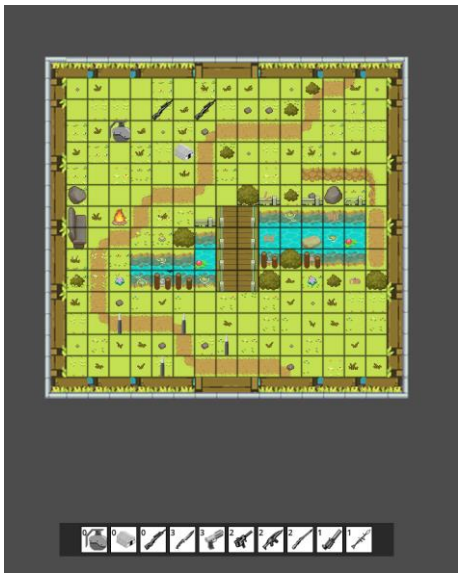
Figure 10
Story Board



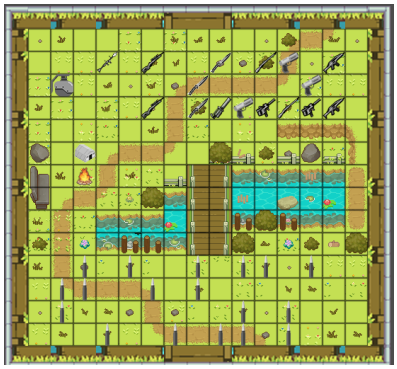
Load Map / Terrain



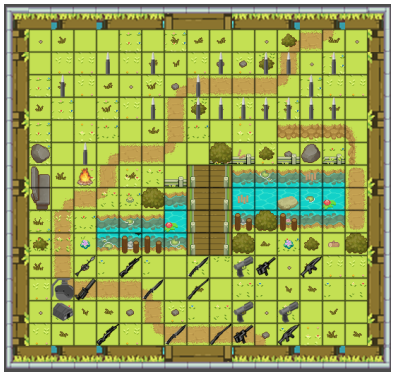
Player 2 Set-up Phase



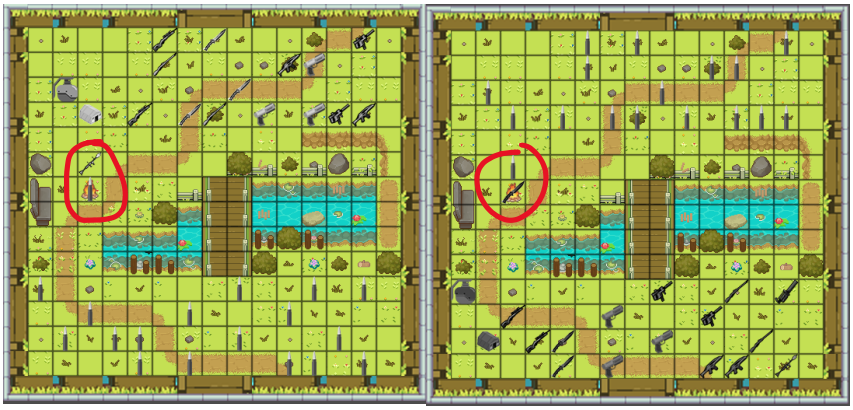
Player 1 Set-up Phase



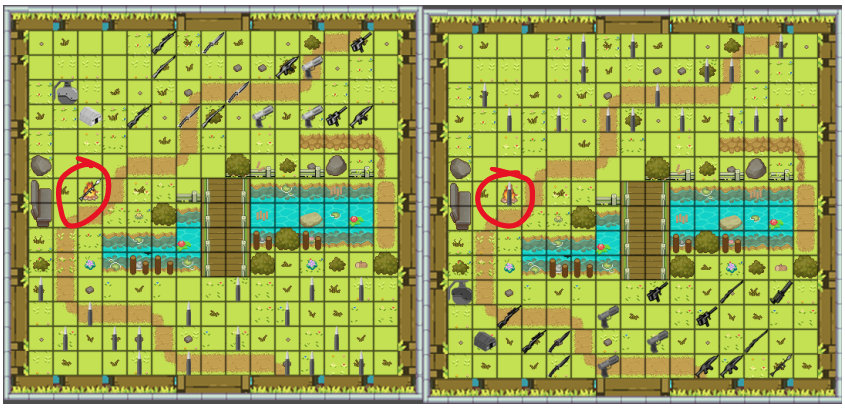
P1 Turn; P2 Units Hidden



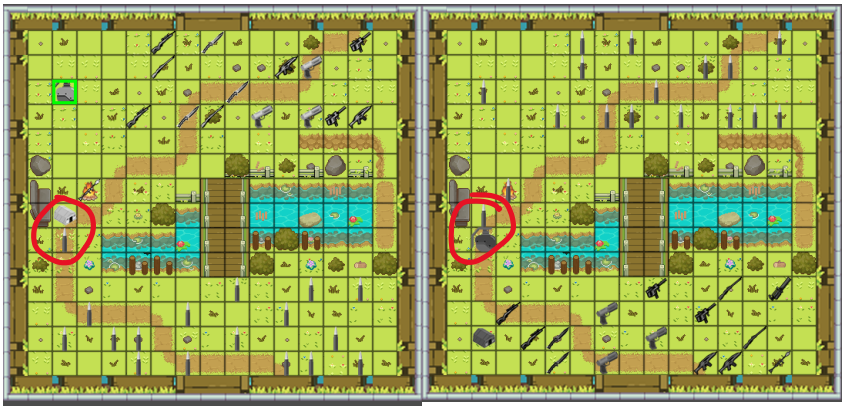
P2 Turn; P1 Units Hidden



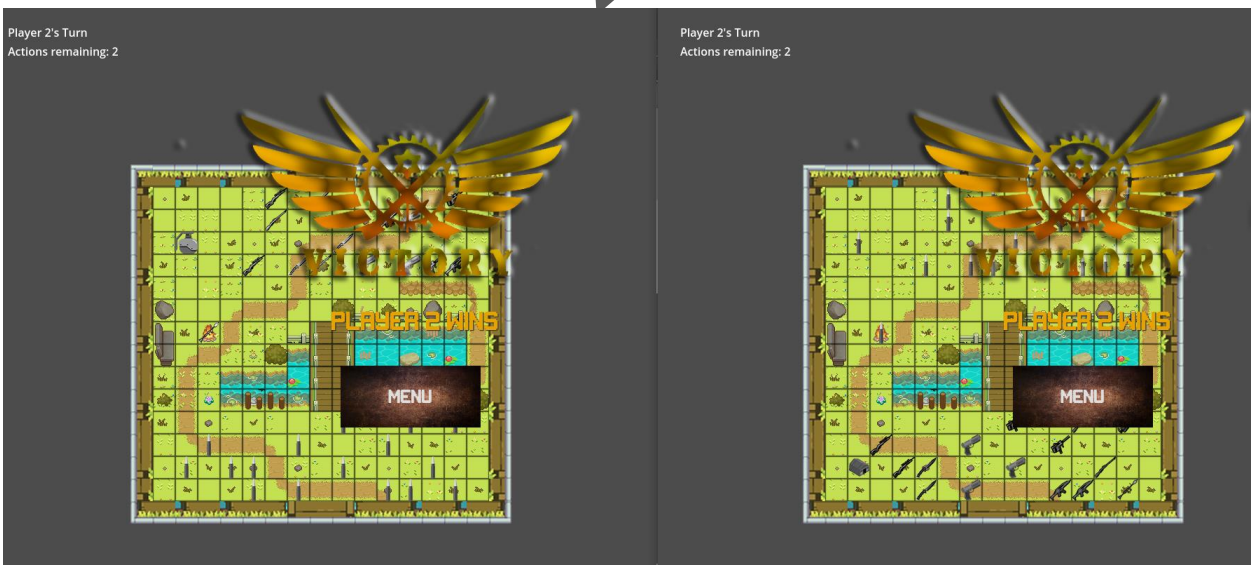
Ensuing Battle (P1 RPG vs P2 Knife)



P1 RPG Wins & Remained on the Board



P1 HQ about to Engage in Battle with P2 Bomber



P2 Wins and Game is Concluded

Wireframe

Figure 11
Wireframes

Figures 11 presents the wireframes for the desired UI Commander. These were the Main Menu, Game type Menu, Settings, Map Selection, the Stage Select, and the Stage wireframes.

Project Development

This section will go over the project's development process and provide an overview of the steps used to create the system in compliance with the design specifications. By aggregating data and relevant studies from across the internet, the researchers were able to determine the issues and goals in the beginning. The researchers identified the issues and proposed a web application system to address them. The researchers will use the Agile Software Development methodology to create this system. The researchers move on to the next stage after each iteration until all requirements are met in this approach, which is based on an iterative procedure. To ensure that the system reflects their vision and receives user feedback, users are involved in the project.

Figure 12
Agile Software Development

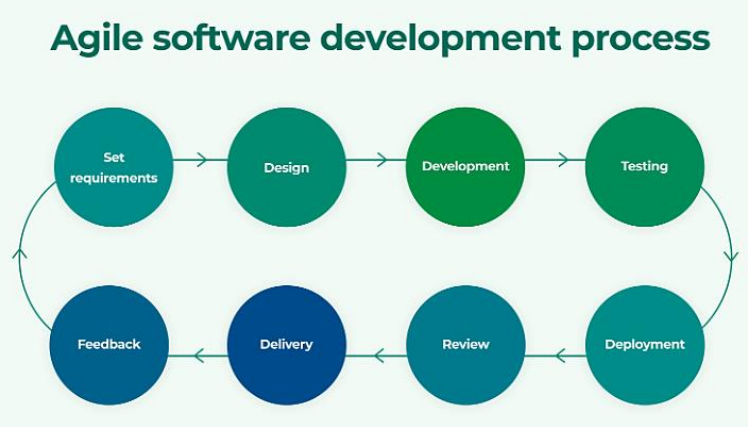


Figure 12 depicts the phases of Agile Software Development; this will serve as a guide for developing the web-based application of the game. Based on the diagram, there are eight (8) phases of iterative approach in developing the system namely: requirements, design, development, testing, deployment, review, delivery, and feedback.

Requirements

The initial step of any game development project is to determine the requirements of the planned game in order to prevent unnecessary scope creep and unexpected delays. For "Commander: A web-based Game of the Generals-inspired game with A.I", the researchers decided to take the existing board game "Game of the Generals" and adds a twist in order to differentiate it from the original inspiration. Features like a new unit system, larger game board with terrain, and an A.I. opponent system were considered among other options in order to set a direction for the project.

Design

For the design phase, the graphical elements of the game are considered here along with the user experience being taken into consideration so that new players wouldn't be confused on how to play the game.

Game Elements

The default plain board in most board games people know has been modified to mimic a terrain with various map-generated blockage, powerups, and traps. These are combined with a unique set of units with varying ranks that will surely make the users familiar even in a short period of time. The game denotes an engaging combination of the original rules and mechanics of the inspiration game with added said twists as well as a more engaging set of units and maps to play into.

Table 2

Commander Unit Hierarchy

Name of Unit	Description	Power
Rocket Launcer	A movable black/white RPG unit model that reflects the look of said weapon.	● Highest in the hierarchy among the guns. Can win against any unit except the Sniper or the Bomber
Machine Gun	A movable black/white gun unit model that reflects the look of said weapon.	● 2nd Highest in the hierarchy among the guns. Can win against any unit except the Sniper, Bomber & Rocket Launcher.
Shotgun	A movable black/white gun unit model that reflects the look of said weapon.	● 3rd Highest in the hierarchy among the guns. Can win against any unit except the Sniper, Bomber & the 2 mentioned ranks higher than this weapon.
Assault Rifle	A movable black/white gun unit model that reflects the look of said weapon.	● The center of the gun-based hierarchy. Can win against any unit except the Sniper, Bomber & the 3 mentioned ranks higher than this weapon.

SMG	A movable black/white gun unit model that reflects the look of said weapon.	● The 3rd to the lowest ranked unit in the gun-based hierarchy. Can only win against units such as the Pistol, Knife and HQ.
Pistol	A movable black/white gun unit model that reflects the look of said weapon.	● The 2nd to the lowest ranked unit in the gun-based hierarchy. Can only win against units such as the Knife and HQ.
Knife	A movable black/white unit model that reflects the look of said weapon.	● The lowest ranked unit in the gun-based hierarchy. Can only win against the HQ as well as it is the only unit that can win against the Sniper.
Sniper	A movable black/white gun unit model that reflects the look of said weapon.	● One of the special case units that lives outside of the hierarchy. It can win against any unit except for the Knife and Bomber.
Bomber	A movable black/white bomb unit model that reflects the look of said weapon.	● One of the special case units that lives outside of the hierarchy. It can win against any unit but eliminates itself as well upon first contact.
HQ	A movable black/white tent unit model that reflects the look of said weapon.	● The only unit with no power. The unit that both players need to protect in order for the game to not conclude.

Unit Sprite Creation

Figure 13
P1 Commander Unit Type Sprite Sheet

Figure 14
P2 Commander Unit Type Sprite Sheet

Figure 15
P1 GG Unit Type Sprite Sheet

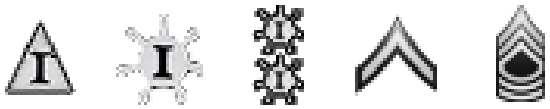
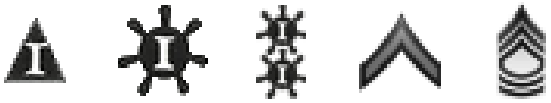


Figure 16
P2 GG Unit Type Sprite Sheet



The figures shown below represent the different guns and ranks in Commander. These sprites were created using the software application Adobe Photoshop. These unit models serve as the playable units of both players across the board in the game. The white colored units are the ones playable for Player 1 while the black colored units are the one playable for Player 2. For a complete view of each unit type for each player, please refer to Appendix I.

Game User Interface

Figure 17
Commander GUI

The following figure presents the User Interface of Commander created in Adobe Photoshop. The game utilized a deserted/old war theme for their buttons and containers. For the tile maps, it has a grass and winter theme.

Game Video, Music and Sound effects

The video effects here served as a background and mechanics were created through Adobe Premier Pro. The specific videos featured in the project are titled 'commander-main-menu ogv, 'desert grass ogv, 'desert winter ogv, 'battle-mechanics ogv, 'cross-movement ogv, 'partially-observed ogv', 'turn-based ogv', and 'winning-condition ogv'. The audio sound effects used in the development of Commander came from. The specific songs featured in the project are titled 'heroic-intro sfx, 'encounter sfx', 'game lose sfx, 'game win sfx', 'menu buttons sfx', 'turn change sfx', and 'unit placement and movement sfx'. The music used in this thesis is accessible and licensable through platforms such as Pixabay.com, Envato.com, and Freesound.org, where the artists have made their work available to the public.

Game Development

This section is dedicated to the code implementations that allow the game to function as intended. The researchers used the Godot Game Engine and Visual Studio as the development tools to write and implement the necessary algorithms for the game.

The following are the scripts that were created for Commander:

- **Board Script** – This script creates the abstracted board that syncs with the TileMap used in the game Engine. It includes the places where the pieces move upon as well as where the tile blocks are placed.
- **Map Loading Script** – This script handles the information of the board map designs that the players choose. It then loads the chosen map for that game.
- **Turn Handler Script** – This script handles declares whose turn it currently is and changes the board to reflect that. It also records the move counter of the game since each player has 6 moves per turn.
- **Player 1 Pieces Script** – This script creates the logic to determine which pieces belong to "Player 1" of the game. It includes which assets the game needs to use for the specific player, how many pieces in total will they use, and the number of pieces with a specific rank they can place.
- **Player 2 Pieces Script** – This script creates the logic to determine which pieces belong to "Player 2" of the game. It includes which assets the game needs to use for the specific player, how many pieces in total will they use, and the number of pieces with a specific rank they can place.
- **Setup Script** – This script handles the setup phase for both players. It allows the placement of the player's pieces that is restricted to the first four rows of their respective sides.
- **Movement Script** – This script handles the logic for moving pieces from one tile to another. It also prevents pieces from moving into tiles that have pieces of the same player along with tiles that are out of bounds.
- **Possible Moves Script** – This script handles the possible moves a player can make with one piece. After collecting the possible moves it then highlights them whenever a player has chosen a piece to move.

- **Rank-Based Eating Script** – This script defines the values of pieces and puts them into specific ranks to compare. It is mainly used to simulate an arbiter in real life by comparing the values of pieces that come into contact with one another and remove the weaker one or both if they are of equal rank.
- **Update Board Script** – This script handles updating the board state as well as piece info for each piece after any movement or battle between pieces.
- **Hide Enemy Board Script** – This script handles concealing the enemy board pieces whenever it is not their turn. This forces players to guess and consider their options when trying to move onto enemy tiles.

Implementation of Setup

Figure 18
Offensive Strategy Setup

```
2316
2317  func _on_offensive_pressed():
2318
2319
2320      if Global.current_player == 1:
2321          var offensive_choice_board = Global.player1_offensive_options[randi_range(0,2)]
2322          for unit in offensive_choice_board:
2323              Global.unit_dict[str(unit)] = offensive_choice_board[unit]
2324              Global.p1_unit_dict[str(unit)] = offensive_choice_board[unit]
2325              piece_type = offensive_choice_board[unit]["Type"]
2326              id_coords = p1_id_dict.get(piece_type, Vector2i(0,0))
2327              set_cell(1, Vector2i(unit), id_coords, Vector2i(0,0))
2328          setup_phase_player1 = false
2329          switch_turn()
2330          hide_enemy_board()
2331
2332      elif Global.current_player == 2:
2333          var offensive_choice_board = Global.player2_offensive_options[randi_range(0,2)]
2334          for unit in offensive_choice_board:
2335              Global.unit_dict[str(unit)] = offensive_choice_board[unit]
2336              Global.p2_unit_dict[str(unit)] = offensive_choice_board[unit]
2337              piece_type = offensive_choice_board[unit]["Type"]
2338              id_coords = p2_id_dict.get(piece_type, Vector2i(0,0))
2339              set_cell(1, Vector2i(unit), id_coords, Vector2i(0,0))
2340          setup_phase_player2 = false
2341          hide_enemy_board()
2342          switch_turn()
```

The code defines a function called on_offensive_pressed(), likely used in a game for offensive moves. It checks the current player's turn, selects a random offensive option for that player, updates unit information on the board, updates visual representation, and progresses the game by switching turns and hiding the opponent's board. This process is repeated for each player's turn.

Figure 19
Defensive Strategy Setup

```
func _on_defensive_pressed():
    if Global.current_player == 1:
        var defensive_choice_board = Global.player1_defensive_options[randi_range(0,2)]
        for unit in defensive_choice_board:
            Global.unit_dict[str(unit)] = defensive_choice_board[unit]
            Global.p1_unit_dict[str(unit)] = defensive_choice_board[unit]
            piece_type = defensive_choice_board[unit]["Type"]
            id_coords = p1_id_dict.get(piece_type, Vector2i(0,0))
            set_cell(1, Vector2i(unit), id_coords, Vector2i(0,0))
        setup_phase_player1 = false
        switch_turn()
        hide_enemy_board()

    elif Global.current_player == 2:
        var defensive_choice_board = Global.player2_defensive_options[randi_range(0,2)]
        for unit in defensive_choice_board:
            Global.unit_dict[str(unit)] = defensive_choice_board[unit]
            Global.p2_unit_dict[str(unit)] = defensive_choice_board[unit]
            piece_type = defensive_choice_board[unit]["Type"]
            id_coords = p2_id_dict.get(piece_type, Vector2i(0,0))
            set_cell(1, Vector2i(unit), id_coords, Vector2i(0,0))
        setup_phase_player2 = false
        hide_enemy_board()
        switch_turn()
```

The code defines a function called on_defensive_pressed(), likely used for defensive moves in a turn-based strategy game. Depending on the current player's turn, it selects a random defensive option, updates unit information on the board, updates visual representation, progresses the game by switching turns, and hides the opponent's board. This process is repeated for each player's turn.

Figure 20
Balanced Strategy Setup

```
2372 func _on_balanced_pressed():
2373     if Global.current_player == 1:
2374         var randposition: Vector2
2375         var piece_array := [bomb1_num, hq1_num, sniper1_num, knife1_num, pistol1_num, smg1_num, rifle1_num, shotgu
2376         for j in range(10):
2377
2378             var counter = piece_array[j]
2379
2380             # randomize valid position
2381             while counter:
2382                 randposition.x = randi_range(0, Global.gridsizeX-1)
2383                 randposition.y = randi_range(0, Global.gridsizeY-1)
2384
2385                 # randomly place pieces
2386
2387                 if str(randposition) not in unit_dict and str(randposition) not in p1_unit_dict and randposition.y
2388                     set_cell(1, randposition, Global.p1_id_dict.values()[j], Vector2i(0,0))
2389                     if j == 0:
2390                         Global.unit_dict[str(randposition)] = {
2391                             "Type": "Bomb",
2392                             "Color": 1,
2393                             "Owner": "Player"
2394                         }
2395                         Global.p1_unit_dict[str(randposition)] = {
2396                             "Type": "Bomb"
2397                         }
2398                         player_vector_array.append(randposition)
2399                     if j == 1:
2409                     if j == 2:
2419                     if j == 3:
```

The code defines a function on_balanced_pressed(), which is used to place game pieces on a grid in a balanced or random manner, likely during the setup phase of a turn-based strategy game. It first checks if it's player 1's turn. If so, it initializes a list for storing random grid positions and

an array of different game pieces. It iterates to place each type of piece on the grid by finding valid random positions within specified bounds. For player 1, pieces are restricted to the bottom half of the grid. The function uses set_cell() to place pieces and updates relevant global dictionaries with the piece details.

Figure 21

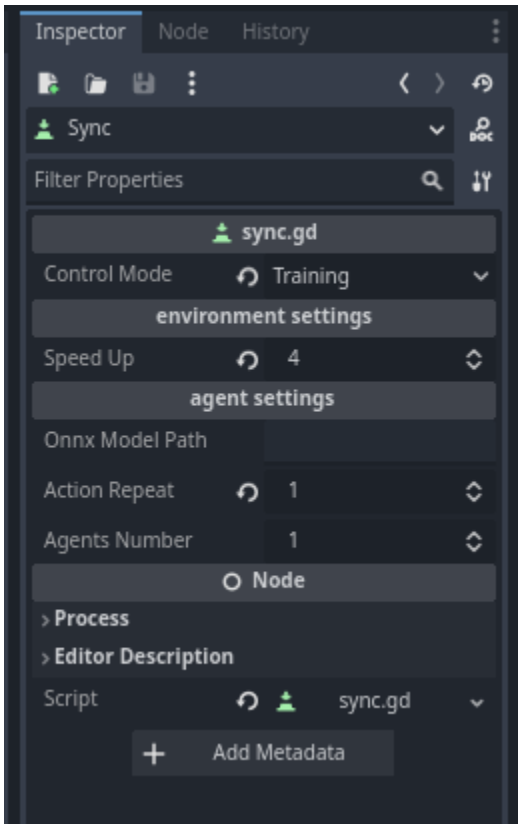
Randomized Strategy Setup

```
2372 func _on_balanced_pressed():
2373     if Global.current_player == 1:
2374         var randposition: Vector2
2375         var piece_array := [bomb1_num, hq1_num, sniper1_num, knife1_num, pistol1_num, smg1_num, rifle1_num, shotgu
2376         for j in range(10):
2377             
2378             var counter = piece_array[j]
2379 
2380             # randomize valid position
2381             while counter:
2382                 randposition.x = randi_range(0, Global.gridsizeX-1)
2383                 randposition.y = randi_range(0, Global.gridsizeY-1)
2384                 
2385                 # randomly place pieces
2386                 
2387                 if str(randposition) not in unit_dict and str(randposition) not in p1_unit_dict and randposition.y
2388                     set_cell(1, randposition, Global.p1_id_dict.values()[j], Vector2i(0,0))
2389                     if j == 0:
2390                         Global.unit_dict[str(randposition)] = {
2391                             "Type": "Bomb",
2392                             "Color": 1,
2393                             "Owner": "Player"
2394                         }
2395                         Global.p1_unit_dict[str(randposition)] = {
2396                             "Type": "Bomb"
2397                         }
2398                         player_vector_array.append(randposition)
2399                     if j == 1:
2409                     if j == 2:
2419                     if j == 3:
```

The player_random_setup() function randomly sets up a player's pieces on a game board in a turn-based strategy game. If it's player 1's turn, the function initializes a list for random positions and an array of various game pieces. It iterates to place each type of piece on the grid by finding valid random positions within specified bounds, ensuring player 1's pieces are restricted to the bottom half of the grid. Valid positions are not already occupied and meet certain criteria. The function places each piece using set_cell(), updates global dictionaries with piece details, and tracks the positions. Similar actions for other piece types are indicated but incomplete.

Implementation of AI

Figure 22
Godotrl-web



Godot RL Agents is an open-source plugin for Godot that allows users to implement machine learning models in the Godot Game Engine. Figure 22 shows the settings that the user can adjust while inside the Godot Game Editor. The inputs to the Neural Network via the Godot RL Plugin are defined by 4 main functions:

Figure 23
Getting user inputs

```
func get_obs() -> Dictionary:
    # inputs:
    # unit count for player and ai
    # coordinates of player pieces
    # coordinates of ai pieces
    # strength of ai pieces
    var env := [
        tile_map.p1_unit_count,    # number of enemy pieces
        tile_map.p2_unit_count,    # number of friendly pieces
    ]
    env = env + tile_map.ai_all_piece_positions
    #print(env)
    return {"obs":env}
```

The "**get_obs()"** function allows the user to set the inputs to the neural network. The inputs must be formatted as a 1-dimensional array of floats or integers to be used as the value of a dictionary.

Figure 24
Assigning rewards

```
func get_reward() -> float:
    # BIG reward for winning the game
    # reward for killing a enemy pieces
    # DONE reward for moving when a move is given
    # BIG penalize for losing
    # DONE penalize selecting dead pieces
    # DONE penalize selecting tiles that arent friendly pieces
    # penalize having a piece killed
    # penalize taking too long to "act" (ai must be kinda offensive)
    return reward
```

The "**get_reward()"** function lets the user set up the reward system of the neural network. The goal of any neural network is to maximize the rewards that it can achieve. This means that the network is incentivized to take and avoid certain actions by rewarding and punishing the network respectively for doing certain things that are defined by the user.

Figure 25
Setting the outputs of the Neural Network

```
func get_action_space() -> Dictionary:
    return {
        # selected tile
        "origin-x-coordinate" : {
            "size": Global.gridsizeX,
            "action_type": "discrete"
        },
        "origin-y-coordinate" : {
            "size": Global.gridsizeY,
            "action_type": "discrete"
        },
        "cardinal" : {
            "size": 4,
            "action_type": "discrete"
        },
    }
```

The "**get_action_space()**" function lets the user set up what the output of the neural network is supposed to be. In our example it returns a dictionary with certain user-defined parameters with the rest of the game code can use to do further processing with.

Figure 26
Output Formatting

```
func set_action(action) -> void:
    raw_action = action
    origin_tile.x = action["origin-x-coordinate"]
    origin_tile.y = action["origin-y-coordinate"]
    cardinal = action["cardinal"]
```

The "**set_action()**" function defines the format of the outputs that it gets from "**get_action_space()**" to user defined variables which can be used by the game code.

In our game, we use these functions to set the inputs as a list of integers holding the player and ai unit counts, unit positions, and ai unit strengths. We then return the reward value from the "**get_reward**" function as the reward variable to be used directly in the code for training purposes. The outputs of the neural network is then set as the coordinate of the tile that it wants to play and the movement direction of the tile. Lastly, outputs are then formatted for access by the game.

Figure 27

AI piece selection and movement

```
if ml_controller.raw_action:
    var ai_origin_tile: Vector2 = ml_controller.origin_tile
    var ai_destination_tile: Vector2
    var direction: int = ml_controller.cardinal + 1

    if direction == 1: # north
        ai_destination_tile = ai_origin_tile + Vector2(0, -1)
    elif direction == 2: # east
        ai_destination_tile = ai_origin_tile + Vector2(1, 0)
    elif direction == 3: # south
        ai_destination_tile = ai_origin_tile + Vector2(0, 1)
    elif direction == 4: # west
        ai_destination_tile = ai_origin_tile + Vector2(-1, 0)

    if str(ai_origin_tile) in p2_unit_dict:

        if xInbounds(ai_destination_tile) and yInbounds(ai_destination_tile) and str(ai_destination_tile) not in p2_unit_dict:
            print("moved from " + str(ai_origin_tile) + " to " + str(ai_destination_tile))
```

```
if str(ai_origin_tile) in p2_unit_dict:

    if xInbounds(ai_destination_tile) and yInbounds(ai_destination_tile) and str(ai_destination_tile) not in p2_unit_dict:
        print("moved from " + str(ai_origin_tile) + " to " + str(ai_destination_tile))

        var temp = p2_unit_dict[str(ai_origin_tile)]
        p2_unit_dict.erase(str(ai_origin_tile))
        p2_unit_dict[str(ai_destination_tile)] = temp
        erase_cell(1, ai_origin_tile)
        set_cell(1, ai_destination_tile, Global.p2_id_dict[temp["Type"]], Vector2i(0,0))

        temp = unit_dict[str(ai_origin_tile)]
        unit_dict.erase(str(ai_origin_tile))
        unit_dict[str(ai_destination_tile)] = temp

        _ai_preprocess_input()

        ml_controller.reward += 1
    else:
        print("movement failed")
        ml_controller.reward -= 1
```

Aside from the functions made to configure the neural network. There is also a script for training the AI in order to select and move its own pieces. It gets the origin tile of a piece and then tries to move it in all 4 directions. After it moves we then get informed where it moved to and then get a reward to give to the controller depending on what happens after the piece has been moved.

Figure 28
Example Training

On the figure above is the statistics when training the AI. It shows important properties such as the total number of iterations it has currently gone through and the time elapsed since start. It also includes the training statistics such as the learning rate, which determines how fast the model learns from the data it has been given. The loss, which is how well the model performs. And one of the more important parts which is the approx._kl property which stands for "Approximate Kullback-Leiber Divergence". This property measures the difference between the old and new policy after an update. A small value here is good because it means that the training is stable and doesn't really deviate from one extreme to another.

Testing

Testing becomes a critical phase, encompassing functional and performance testing within the Godot engine. The researchers ensure that the AI opponents display strategic intelligence, multiplayer interactions are seamless, and the web-based gameplay operates smoothly across various browsers. Godot's debugging tools aid in identifying and addressing any issues to guarantee a polished gaming experience.

Deployment

Following completion of the testing phase, the game will be exported to a website-ready format and hosted on a website hosting service in preparation for the final step.

Review

The researchers will reevaluate the commercial readiness of the application. The researchers will develop answers for the problems that were present in previous versions.

Delivery

The program should now be re-delivered to users after completing all prior development steps.

Feedback

The researchers will gather user feedback that will be recorded for improvements to the game in a continuous feedback cycle in order to make the gameplay better and fix bugs when they pop up. This also allows for the addition of new features and mechanics in order to keep the gameplay fresh and relevant.

Operation and Testing Procedure

The following procedure will be followed in order to operate the Commander Application as well as to test if all the intended system features are working properly

1. Start up the Commander Game
2. If inexperienced, familiarize yourself with the rules as well as the new types of pieces used.
3. Select a Map and Piece Design that you want to use.
4. Play several games to completion to test the Gameplay loop as well as the AI.

Table 3

Operation and Testing Procedure

System Function	Procedure	Expected Output
Learning	Click on the Info button Information about the game and the new pieces and their counterparts in Game of the Generals will then be presented. There is also info on new pieces and special cases on how they are used.	The user should be able to familiarize themselves with the game and how to play it. The application should handle all scenarios without crashing
Map Selection	Before playing, the user can choose which map they would like to play on	The user should be able to choose which map they want without encountering any kind of error or failure to load
Setup Phase	Before starting the actual match, the users need to set up their respective pieces on their sides of the Map	The user should be able to put down and place their pieces without encountering any bugs
Gameplay	The user can initiate a game between them and another actual person, or an AI to act as their opponent. They will then proceed as normal until one player wins.	The users should be able to move their pieces properly and be able to challenge enemy pieces without any error. The game should also end once one or both of the HQ's have fallen.

A.I powered Opponent	When the player chooses the "Play with Computer" option, the application will employ an AI to the second player position in order to battle against the user.	The user should be able to play against the AI without the AI encountering any errors on producing which moves to use or errors on getting and placing its own pieces.

Relevant Rules & Mechanics

Some of the following rules & mechanics from the reference game "Game of the Generals (GG) will be retained in the creation of this study's application:

1. Turn-based; Players move alternately.
2. Partially Observed; a player won't be able to see the ranks of his/her opponent's units but only their position on the playing field.
3. Unit ranking system; GG: Private – 5* General; CM: Knife – Rocket Launcher
4. A move consists of pushing a piece to an adjacent square, either forward, backward or sideward. A diagonal move or a move of more than one square is illegal.
5. A battle will ensue when one unit is moved to the square of an opposing unit, the unit with the higher rank will conquer/remain the said square. (Both units are eliminated if both has the same rank).
6. An arbiter is needed to judge the unit battles.

Differences in Mechanics

Table 4

Commander vs GG Mechanics

Games of the Generals	Commander
9x8 Gameboard	15x14 Gameboard
21 total pieces per player	18 total pieces per player
1 action per turn	6 actions per turn
Military Ranking System	Gun-based Ranking
Standard Map	Land-Based Maps (Squares may contain obstacles relating to map theme)
9x3 Deployment Field	15 x 4 Deployment Field
Deployment Phase – Unit Deployment	Deployment Phase – Unit & Bomber Deployment
Game ends = flag eliminated / flag reached the end of the other side.	Game ends = HQ eliminated
N/A	Piece Hierarchy Legend
N/A	Setup type per player (random, offensive, defensive, balance)

Differences in Unit Rankings

Table 5

Commander vs GG Differences in Unit Rankings

Games of the Generals	# of Pieces	Commander	# of Pieces
5 STAR GENERAL	1	-	0
4 STAR GENERAL	1	-	0
3 STAR GENERAL	1	-	0
2 STAR GENERAL	1	-	0
1 STAR GENERAL	1	-	0
COLONEL	1	-	0
LIEUTENANT COLONEL	1	ROCKET LAUNCHER	1
MAJOR	1	MACHINE GUN	1
CAPTAIN	1	SHOTGUN	2
1ST LIEUTENANT	1	RIFLE	2
2ND LIEUTENANT	1	SUBMACHINE GUN	2
SERGEANT	1	PISTOL	3
PRIVATE	6	KNIFE	3
SPY	2	SNIPER	2
FLAG	1	HQ	1
-	-	SUICIDE BOMBER	1

Evaluation Procedure

The evaluation that will be used to assess the acceptability of the application will be based on the ISO 25010 which the "Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality modes".

The following procedure will be conducted to assess the acceptability of the developed application:

The researchers will request a total of 30 respondents which is composed of 15 TUP students without technical backgrounds and the other half consisting of individuals from outside the TUP community with technical expertise to give their ratings.

The researchers will demonstrate to the respondents on how to operate the application. The respondents will now be asked to try using the application by themselves to know if they can navigate it with ease.

The respondents will be invited to evaluate the application based on the given evaluation sheets using a 4-point Satisfaction scale.

The accomplished assessment sheets will be processed, and the information gathered form the evaluation will be tabulated to determine the mean ratings.

Table 6

4-Point Likert Scale.

Numerical Rating	Descriptive Rating
4	Highly Acceptable
3	Very Acceptable
2	Fairly Acceptable
1	Not Acceptable

Table 6 shows the equivalent rating and interpretation of the evaluation using the 4-Point Likert Scale. This will be used to evaluate the score that the application got from the evaluation of the respondents.

Table 7

4-Point Likert Scale.

The Range of Weighted Mean Ratings and its Qualitative Representation

Numerical Rating	Descriptive Rating
3.4 - 4.0	Highly Acceptable
2.6 - 3.3	Very Acceptable
1.8 - 2.5	Fairly Acceptable
< 1.7	Not Acceptable

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter presents the project description, project structure, project test results, project capabilities and limitations, and project evaluation results of the study.

Project Description

Commander is a 2D board game which makes use of existing ruleset of the base game ‘game of the generals’ with added rules and features as well as to develop an A.I. adversary that can realistically simulate an actual human opponent. The software application is currently compatible with devices running the Windows operating system. Commander has been developed using the GDscript programming language and Godot game engine. Additionally, other tools and platforms such as Adobe Photoshop are utilized for sprite creation and user interface, Adobe Premier Pro for background and mechanics videos.

Project Structure

Figure 29
Main Menu Screen

Figure 29 depicts the Commander’s Main Menu screen, from which the user can select Play, View the Mechanics (which serves as the game's primary tutorial of the Game), Settings, and Exit the game.

Figure 30
Settings Screen

Figure 30 shows the Commander's settings screen, from which the user can adjust the volume of either the master music, or just the sound effects. In this screen, the user can also personalize which unit set the game offers to play.

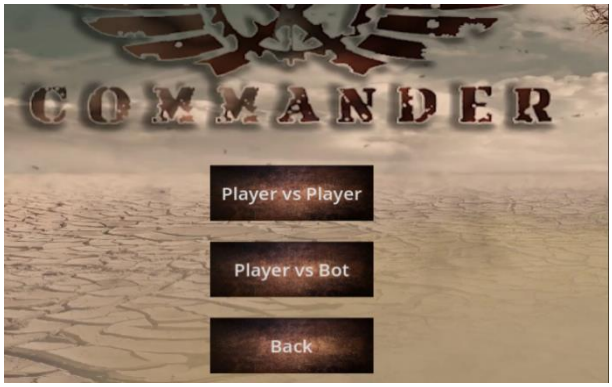
Figure 31
Mechanics Screen

Figure 31 portrays the Commander's mechanics screen, from which the user can review the overall rules and mechanics of the game. This screen can also familiarize the user with the game's unit hierarchy system.

Figure 32
Map Selection Screen

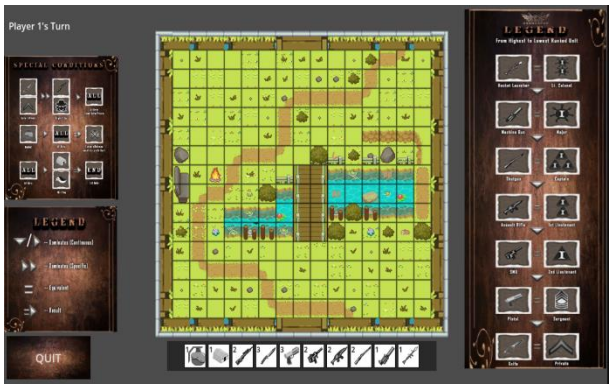
The map selection screen, seen in Figure 32, will appear when the player presses the "Play" button on the Main Menu screen. The game's multiple map options are shown on this screen.

Figure 33
Game Mode Selection Screen



The game mode selection screen being illustrated in Figure 33, will appear upon clicking a map to play on. This screen offers the user 2 game mode selection, one is to play with another player, and another is to play against an AI bot. Lastly, the third button “Back” is placed for user convenience.

Figure 34
Gameplay Screen



The gameplay screen as shown in Figure 34, will be the screen for the user/s to interact with the main aspects of the game. Here lies the main board, a 15x14 board with 2 sides where player/s will place their units before the main game starts. This is also the screen where the Legend/Unit hierarchy is shown for user convenience and familiarity.

Figure 35
Game End – Victory Screen

Figure 35 portrays the Commander's victory screen, triggered when either of the player's HQ unit is eliminated from the board thus concluding the game. This screen also has the Menu button for user convenience if ever the user wants to start another game.

Figure 36
Game End – Draw Screen

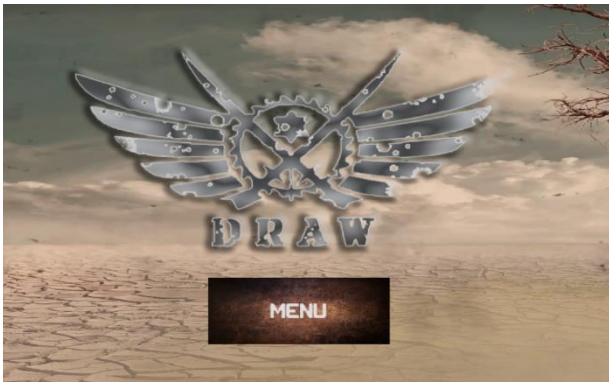


Figure 36 portrays the Commander's draw screen, triggered when both of the player's HQ unit collide and eliminated each other thus concluding the game. This screen also has the Menu button for user convenience if ever the user wants to start another game.

Table 8
Operations and Testing Results

Test Cases	Steps Undertaken	Observed Results
Access the Map Selection scene	1. Start the game and press Play	The user is successfully loaded into the Map Selection scene

Access the Options scene	1. Start the game and press Options	The user is successfully loaded into the Options scene
Changing Volume or Piece Design	1. Start the game and press Options 2. Interact with either the Volume Slider or the Toggle for Piece Design	The Volume is adjusted correctly and the chosen Piece Design is kept when starting a game
Completing the Setup Phase	1. Start the game and press Play 2. Choose a Map to play on 3. Interact with your hot bar and press on the appropriate tiles to place your units	The user can successfully place their pieces in the initially chosen tiles. The pieces are then hidden from the enemy depending on which turn it is.
Selecting and Moving pieces	1. Start the game and press Play 2. Choose a Map to play on 3. Interact with your hot bar and press on the appropriate tiles to place your units 4. If you are the current player, try to select your pieces by clicking on it and then clicking again to place it down	The user is able to successfully select the pieces that were setup and move them to another tile
Rank-Based piece eating	1. Start the game and press Play 2. Choose a Map to play on 3. Interact with your hot bar and press on the appropriate tiles to place your units	The user is able to put their pieces into tiles with enemies and the script for the arbiter is working as intended for the rules of the game.

	4. If you are the current player, try to select your pieces by clicking on it and then clicking again to place it down on where there is an enemy piece	
Hiding enemy board	1. Start the game and press Play 2. Choose a Map to play on 3. Interact with your hot bar and press on the appropriate tiles to place your units 4. If you are the current player, try to select your pieces by clicking on it and then clicking again to place it down 5. After moving enough times to end your turn, observe the board to see if your pieces get concealed once it is the enemy's turn	The user's units are properly obstructed from the enemy's perspective and replaced by an asset to hide their ranks.
Victory or Draw Screen	1. Start the game and press Play 2. Choose a Map to play on 3. Interact with your hot bar and press on the appropriate tiles to place your units	The user is successfully loaded into the Victory / Draw game scene depending on the result of the game.

	4. If you are the current player, try to select your pieces by clicking on it and then clicking again to place it down on where there is an enemy piece 5. Play the game to its completion by eliminating the opponents HQ	

Summary of Results

Table 9

Summary of Results

Characteristic	Mean	Evaluation
Functional Suitability		
Functional Completeness	3.4	Highly Acceptable
Functional Correctness	3.2	Very Acceptable
Functional Appropriateness	3.73	Highly Acceptable
Mean Rating	3.44	Highly Acceptable
Performance Efficiency		
Time Behavior	3.37	Very Acceptable
Resource Utilization	3.07	Very Acceptable
Capacity	3.37	Very Acceptable
Mean Rating	3.27	Very Acceptable
Usability		
Learnability	3.57	Highly Acceptable
Operability	3.27	Very Acceptable

Mean Rating	3.42	Highly Acceptable
Reliability		
Availability	3.57	Highly Acceptable
Fault Tolerance	3.07	Very Acceptable
Mean Rating	3.32	Very Acceptable
Maintainability		
Reusability	3.50	Highly Acceptable
Modifiability	3.30	Highly Acceptable
Testability	3.4	Highly Acceptable
Mean Rating	3.4	Highly Acceptable
Overall Mean Rating	3.37	Very Acceptable

The evaluation of various characteristics for the system has shown positive results. For Functional Suitability, the mean rating for Functional Completeness is 3.4, indicating a "Highly Acceptable" level of satisfaction. Functional Correctness scored 3.2, deemed "Very Acceptable," while Functional Appropriateness achieved a higher score of 3.73, also considered "Highly Acceptable." The overall mean rating for Functional Suitability stands at 3.44, falling under the "Highly Acceptable" category.

Moving on to Performance Efficiency, the system's Time Behavior and Capacity both received a rating of 3.37, and Resource Utilization was rated at 3.07. All three aspects are classified as "Very Acceptable," with a collective mean rating of 3.27. In terms of Usability, the system's Learnability received a rating of 3.57, indicating it is "Highly Acceptable," while Operability was rated at 3.27, falling under "Very Acceptable." The mean rating for Usability is 3.42, placing it in the "Highly Acceptable" range. For Reliability, the Availability aspect scored 3.57, categorized as "Highly Acceptable," and Fault Tolerance was rated at 3.07, considered "Very Acceptable." The mean rating for Reliability is 3.32, falling under the "Very Acceptable" category. Lastly, in the Maintainability characteristic, Reusability was rated at 3.50, Modifiability at 3.30, and Testability at 3.4. All three ratings are classified as "Highly Acceptable," leading to an overall mean rating of 3.4 for Maintainability.

In conclusion, the Overall Mean Rating of the system across all evaluated characteristics is 3.37, which is categorized as "Very Acceptable." This indicates a generally high level of satisfaction with the system's functional suitability, performance efficiency, usability, reliability, and maintainability.

Chapter 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the summary of findings, conclusions, and recommendations based on the results of the evaluation comments and suggestions.

Summary of Findings

The study yielded the following results based on examinations and assessments carried out to gauge the performance capacity of the application:

The software was developed according to the planned design and specifications. The program allowed users to engage in a unique version of the classic board game, Game of the Generals, that included features such as map and piece design that separated itself from the original. Extensive testing was conducted on the application, which successfully met the criteria and principles outlined in ISO 25010. This indicated that the application effortlessly utilized and adapted to devices chosen by the users.

After analyzing the data collected during the project evaluation, the study received an overall rating of 3.37, classified as "Very Acceptable" based on the adjectival rating. Specifically, the study achieved the following:

- In terms of the suitability of functionality, the software received a Highly Acceptable rating, indicating that the objectives were successfully achieved in terms of its intended purpose and function.
- In terms of the performance efficiency, the software received a Very Acceptable rating, indicating that the application responds and executes its functions within an above average acceptable response time.
- In terms of portability, the software received a Highly Acceptable rating, which proves that the application can easily be installed and played on devices with different screen resolutions without any errors, as long as they meet the minimum requirements.

Conclusions

In consideration of the objectives of the study, the results of the testing, and the evaluation undertaken, the following conclusions can be drawn:

1. Commander board game is effectively designed and created with the following features:
    a. The sprite models in the game are based on the more common guns that you can see in wars, although there is also an option to use the classic design that was used in Game of the Generals in order to give an option of familiarity.
    b. Players have the ability to place and move their pieces within the board within their turn, which when moved to a tile with an enemy piece will then initiate a comparison between the Player's piece and the enemy's hidden piece.
    c. The partially observable nature of the board makes it so that the enemy's pieces being hidden makes playing and thinking of moves more strategic.
    d. The game features two different map layouts, each with their own unique design and tile set.
    e. The game has different setup a player can use for its initial placement which are randomize, offensive, defensive, and balanced.
2. The game was developed using Godot Game Engine, Visual Studio, Adobe Premiere Pro, & Adobe Photoshop.
3. The game underwent successful testing to evaluate its functionality, performance, and portability, which demonstrated its high quality.
4. Commander was comprehensively evaluated using the criteria of ISO 25010 and received an overall weighted mean of 3.37 with a descriptive rating of Very Acceptable.

Recommendations

The following suggestions are provided for future enhancement of the game, based on the findings and conclusions of the study:

1. Improve the gameplay loop by adding more random factors in play such as powerups that appear in random tiles, or more terrain-based mechanics that affect movement.

2. Add a functionality to pause and even leave while in the middle of an ongoing match. This will improve the quality of life of the game since the players don’t need to play a match to its finish to be able to leave the game.
3. Implement more maps that have terrain variety as well as possible mechanics unique to certain maps that can either benefit or harm the player.
4. Implement different game modes such as Blitz mode and Capture mode. Blitz Mode introduces a fast-paced dynamic with each player having 10 minutes total for all moves, adding a time management element. Capture Mode focuses on capturing a specific number of enemy pieces, with the game ending when a player captures 10 pieces, regardless of the flag's status. These modes diversify gameplay, appeal to different player preferences, and add new strategic dimensions.
5. Defensive type of A.I

References

6Wresearch. (n.d.). Philippines board games market (2020 - 2026): Trends, outlook & forecast. Philippines Board Games Market (2020 - 2026) | Trends, Outlook & Forecast. https://www.6wresearch.com/industry-report/philippines-board-games-market-2020-2026

Anderson, E. F., McLoughlin, L., Liarokapis, F., Peters, C., Petridis, P., & de Freitas, S. (2018). Developing serious games for cultural heritage: A state-of-the-art review. Virtual Reality, 22(2), 101-132.

Ansong, E., & Ong, K. C. (2020). Cross-platform game development using HTML5 and JavaScript. In Proceedings of the International Conference on Computational Science and Its Applications (pp. 485-498). Springer.

Chang, A. (2020, December 15). How online gaming has become a social lifeline. BBC Worklife. https://www.bbc.com/worklife/article/20201215-how-online-gaming-has-become-a-social-lifeline

Chen, S. Y., & Pan, Z. (2013). Integrating game-based learning into software engineering education. IEEE Transactions on Education, 56(4), 454-459.

Costa, L. M., Drachen, A., Souza, F. C., & Xexéo, G. (2023). Artificial Intelligence in MOBA games: A multivocal literature mapping. IEEE Transactions on Games, 1–23. https://doi.org/10.1109/tg.2023.3282157

Curran, K., & George, C. (2012). The future of web and mobile game development. International Journal of Cloud Computing and Services Science (IJ-CLOSER), 1(1). https://doi.org/10.11591/closer.v1i1.233

Ehrenfeld, L. (2022, January). Benefits of board games for children and their families. Child Development Clinic. https://www.childdevelopmentclinic.com.au/benefits-of-board-games-for-children-and-their-families.html

Haro, J., Perez-Sorrosal, F., Conde, C., & Ortega, M. (2019). Enhancing web-based multiplayer games through the integration of WebSockets. In Proceedings of the 14th International Conference on the Foundations of Digital Games (pp. 1-10).

J. Zhang, H. Li, Y. Teng, R. Zhang, Q. Chen and G. Chen, "Research on the Application of Artificial Intelligence in Games," 2022 9th International Conference on Digital Home (ICDH), Guangzhou, China, 2022, pp. 207-212, doi: 10.1109/ICDH57206.2022.00039.

Jiang, C. (2023). The application of artificial intelligence in board games. Proceedings of the 3rd International Conference on Signal Processing and Machine Learning. doi:10.54254/2755-2721/4/20230497

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, 37(2), 83-137.

Ma, Y., & Wang, X. (2017). Challenges and opportunities of web-based 3D game development. Multimedia Tools and Applications, 76(2), 1809-1833.

Malvar.net (2023). Miguel Malvar, a Farmer, Family man and Patriot. (n.d.). http://malvar.net/game-of-the-generals

Mawkins Entertainment. (2020, November 19). Game of the Generals Mobile (Version 1.0) [Mobile application software]. Google Play Store. https://play.google.com/store/apps/details?id=com.mawkins.gogmobile&hl=en&gl=US

Maroney, K. (2001). My entire waking life. The Games Journal [Online]. Retrieved October 25, 2009, from http:///www.thegamesjournal.com/

MetaEngine. (2023, November 24). What is a gaming community?23 And how to build one. MetaEngine. https://www.metaengine.gg/blog/what-is-a-gaming-community-and-how-to-build-one

Müller, L., & Wufka, M. (2018). Towards a comprehensive classification of game engines. Entertainment Computing, 27, 11-31.

Mortensen, E. (2017, June 5). Game of the generals (aka Salpakan) review and rules. Geeky Hobbies. https://www.geekyhobbies.com/game-of-the-generals-aka-salpakan-review-and-rules/

Noda, S., Shirotsuki, K., & Nakao, M. (2019). The effectiveness of intervention with Board Games: A Systematic Review. BioPsychoSocial Medicine, 13(1). https://doi.org/10.1186/s13030-019-0164-1

O'Connor, E., Longman, H., White, K., & Obst, P. (2015). Sense of community, social identity and social support among players of Massively Multiplayer Online Games (MMOGs): A

qualitative analysis1. Journal of Community and Applied Social Psychology, 25(6), 459-4732.

Paciente, D. R. (2023, October 13). Game of the generals - hobbylark. https://hobbylark.com/board-games/how-to-play-game-of-the-generals

Pasola, S. H. Jr. (1973). Game of the Generals. [Board Game]. Philippines.

Rajković, A. I., Ružić, M. S., & Ljujić, B. (2017). Board Games as Educational Media:Creating and Playing Board Games for Acquiring Knowledge of History. International Association for Research on Textbooks and Educational Media, 11(2). https://doi.org/10.21344/iartem.v11i2.582

Samarasinghe, D., Barlow, M., Lakshika, E., Lynar, T., Moustafa, N., Townsend, T., & Turnbull, B. (2021). A data driven review of board game design and interactions of their mechanics. IEEE Access, 9, 114051–114069. https://doi.org/10.1109/access.2021.3103198

Schaul, T., Togelius, J., & Schmidhuber, J. (2011). Measuring Intelligence through Games. ResearchGate. Retrieved from https://www.researchgate.net/publication/51935612_Measuring_Intelligence_through_Games

Schultheiss, D., Bowman, N. D., & Schumann, C. (2008)1. Community vs. soloplaying in multiplayer Internet games2. In Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology (pp. 1-8). ACM.

Stemmle, C. (2023, March 9). 8 mental benefits of playing board games. Happier Human. https://www.happierhuman.com/benefits-board-games/

Studocu. (2021). The game of the generals - this game is considered as an important educational tool.. https://www.studocu.com/ph/document/university-of-the-philippines system/management-accounting/the-game-of-the-generals/20866612

Studocu. (2022). Game of the generals - individual and dual sports introduction the game of Generals. https://www.studocu.com/ph/document/saint-louis-college/physical-education/game-of-the-generals/42962167

Suleimenov, I. E., Vitulyova, Y. S., Bakirov, A. S., & Gabrielyan, O. A. (2020). Artificial Intelligence. Proceedings of the 2020 6th International Conference on Computer and Technology Applications. https://doi.org/10.1145/3397125.3397141

Tang, C., Wang, Z., Sima, X., & Zhang, L. (2020). Research on Artificial Intelligence algorithm and its application in games. 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM). https://doi.org/10.1109/aiam50918.2020.00085

The NPD Group. (2020, November 30). Across all age groups, U.S. consumers are investing more of their entertainment participation. https://www.npd.com/news/press-releases/2020/across-all-age-groups-us-consumers-are-investing-more-of-their-entertainment-participation/

The Times Journal. (2015). THE GAME OF THE GENERALS' STORY. Game of the Generals. https://ggsalpakan.weebly.com/history.html

Treher, E. N. (2011). Learning with Board Games. The Learning Key, Inc.

Wu, B., & Wang, A. I. (2012). A guideline for game development-based learning: A literature review. International Journal of Computer Games Technology, 2012, 1–20. https://doi.org/10.1155/2012/103710

Appendix A

Budgetary Requirements

To successfully develop Commander, the researchers of the study will be using the following items and their respective prices:

Item	Purpose	Budget
Godot Engine	This software program serves as the backbone in creating the scripts, assets and tile maps of the board game	Free
Adobe Photoshop	This software program serves as a tool for creating sprite images, background and button textures.	Free
Adobe Premier Pro	This software program serves as the tool for designing and creating background videos	Free
Google Drive	This cloud software serves as a cache to store assets created by the developers to be used for collaboration purposes.	Free
Visual Studio IDE	This integrated development environment serves as a tool for creating the scripts and to edit specific functions of the assets created in Godot Game Engine.	Free
MS Word	This tool serves as to create diagrams and tables for our research	Free
Discord	This serves as the communication line for in order for the developers to collaborate in meeting sessions.	Free
Github	This serves as the tool for creating the research repo for us to collaborate and push changes to the code that anyone can pull/access and edit it. Technically this, provides the developers to access the file all at the same time.	Free
	TOTAL	Free

Appendix B

Gantt Chart

Gantt Chart

Legend:	
Planned Activity	
Activity Started	
Activity Completed	

Activity	November				December			
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
Group Brainstorming								
CHAPTER 1								
CHAPTER 2								
Game Platform Discussion								
CHAPTER 3								
Finalizing Proposal Papers								
TOPIC PROPOSAL DEFENSE								

Activity	January				February			
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
Game Output Discussion								
Game Sprites								
Game Story Board								
Front End UI Planning								
CHAPTER 1 Revisions								
CHAPTER 2 Revisions								

Activity	March				April			
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
CHAPTER 3 Revisions								
Godot Game Engine UI								
Back-End Development								
AI Opponent Development								

Initial Data Collection								
CHAPTER 4								

Activity	**May**				**June**			
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
UI Development								
Backend Development								
Project Evaluation								
CHAPTER 5								
Debugging and Refactoring								
FINAL DEFENSE								
Future Project Recommendations								
Final Research Submission								

Appendix C

Software Evaluation Instrument of ISO

25010 Commander Board Game

Name:

_____

Instruction: Please evaluate the software material using the given scale and placing a checkmark (✓) under the corresponding numerical rating:

Numerical Rating and Equivalent

4 – Highly Acceptable **3** – Very Acceptable

2 – Acceptable **1** – Not Acceptable

Characteristics	Rating			
	4	3	2	1
A. Functional Suitability	(Highly Acceptable < -> Not Acceptable)			
1. Functional Completeness. The software's functionalities cover all specified tasks and user objectives				
2. Functional Correctness. The software provides results that are accurate based on the industry standard applications and mathematics.				
3. Functional Appropriateness. The software's functionalities can be used for accomplishment of specified tasks and objectives.				
B. Performance Efficiency	(Highly Acceptable < -> Not Acceptable)			
1. Time-Behavior. Time taken for the software to process data and give responses to its users.				
2. Resource Utilization.				

Software utilization and management of resources when performing its functions.				
3. Capacity. Maximum limitations of the software are enough to perform its functions.				
C. Usability	(Highly Acceptable < -> Not Acceptable)			
1. Learnability. Degree to which the user can learn to operate the software with efficiency and effectiveness.				
2. Operability. Software design that affects the easiness of operating the system.				
D. Reliability	(Highly Acceptable < -> Not Acceptable)			
1. Availability. Software is accessible when required for use.				
2. Fault Tolerance. The system operates as intended despite the presence of hardware or software faults.				
E. Maintainability	(Highly Acceptable < -> Not Acceptable)			
1. Reusability. The software can be used to generate results that can be reused for other projects.				
2. Modifiability. The software can be modified efficiently without introducing defects or degrading existing product quality.				
3. Testability. Software's effectiveness and efficiency can easily be tested through different test criteria				
Comments/Suggestions _____ _____ _____ _____				

Appendix D

Communication Letter

We, the researchers: Jerald C. Casimsiman, Mhari Allen E. Limin, Gabriel C. Maraon, and Christian Noel C. Nacario from the College of Science at the Technological University of the Philippines, are presently conducting a study entitled, "Commander: A Web-Based Game Of The Generals-Inspired Game With A.I". The main objective of the study is to develop a 2D board game based on the Philippine hit game, Game of the Generals. With that said, we would like to ask for your consent to be a part of our testing and evaluation of the game application. I do hereby acknowledge, consent, and agree to all the following terms and conditions:

- I declare that in signing this form, I am agreeing to and giving permission to be part of the testing and evaluation of the game application.
- I hereby attest that my involvement in this activity is voluntary and that I have read (or have read to me) this release, understand, and sign it voluntarily.
- I may revoke this authorization at any time granted.

We assure the complete participant confidentiality that their personal information would be kept private.

Conforme:

_____ _____

Researcher, Group Leader Name and Signature

Noted by:

Dolores Montesines

_____

Research Adviser

Appendix E

Sprite Models

Appendix F

Certificate of Similarity Index Using Turnitin from URDS

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 711 | Fax No. +632-521-4063 Email: urds@tup.edu.ph | Website: www.tup.edu.ph	Index No.	REF-URD-INT-CSI
		Issue No.	01
		Revision No.	01
		Date	04132021
VRE-URD	**CERTIFICATE OF SIMILARITY INDEX USING TURNITIN**	Page	1 / 2
		QAC No.	CC-04132021

This is to certify that the manuscript entitled

"COMMANDER: A WEB-BASED GAME OF THE GENERALS-INSPIRED GAME WITH A.I"

authored by

Casimsiman, Jerald C.
Limin, Mhari Allen E.
Maraon, Gabriel C.
Nacario, Christian Noel C.

has underwent plagiarism similarity check on June 14, 2024, using Turnitin Software with generated similarity index of 7%

Processed by:

DENNIS J. TABUCOL
URDS Staff

Certified correct by:

FRANCISCO D. ESPONILLA II, LPT, Ed.D.
Director, University Research and Development Services

Transaction ID	
Signature	

Appendix G

Thesis Grammarian Certification

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 608| Fax No. +632-8521-4063 Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	
		Revision No.	
		Effectivity Date	
VAA-COS	**THESIS GRAMMARIAN CERTIFICATION**	Page	

This is to certify that the thesis entitled,

COMMANDER: A WEB-BASED GAME OF THE GENERALS-INSPIRED GAME WITH A.I

authored by

Casimsiman, Jerald C.
Limin, Mhari Allen E.
Maraon, Gabriel C.
Nacario, Christian Noel C.

has undergone editing and proofreading by the undersigned.

This Certification is being issued upon the request of Casimsiman, Jerald C., Limin, Mhari Allen E., Maraon, Gabriel C., and Nacario, Christian Noel C. for whatever purposes it may serve them.

MS. FRANZE NAVARRO OROCEO
Grammarian

Technological University of the Philippines

June 12, 2024

Transaction ID	
Signature	

Researcher's Profile

Researchers	Name	Address	Contact Number	Email
	Jerald C. Casimsiman	Blk2 Lot17 Sambakai, Kalayaan, Pasay City	0932 719 0043	jerald.casimsiman@tup.edu.ph
	Mhari Allen E. Limin	573 M. dela Cruz St. Pasay City	0995 976 4691	mhariallen.limin@tup.edu.ph
	Gabriel C. Maraon	Block 10 Lot 21, Phase 5, Santa Barbara Villas 1, Silangan, San Mateo, Rizal, 1850	0930 160 1200	gabriel.maraon@tup.edu.ph
	Christian Noel C. Nacario	1812 Tramo Street Pasay City	0908 466 0212	christiannoel.nacario@tup.edu.ph