

**GAME SIMULATION FOR ALTERNATIVE LEARNING IN BASIC PYTHON FOR
CS AND IT STUDENTS OF CAVITE STATE UNIVERSITY - IMUS CAMPUS**

An Undergraduate Thesis
Submitted to the Faculty of the
Department of Computer Studies
Cavite State University
Imus Campus, Cavite

In partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

**GARGAR, GLENN ACE LIMWELL B.
GERVACIO, RONNETH JOY V.
TOPACIO, KHAIZTER VASH N.**
July 2023

BIOGRAPHICAL DATA

Glenn Ace Limwell B. Gargar was born on January 18, 2002, and is currently residing in Greenside Homes, Imus, Cavite. He is the youngest child among the three siblings of Jeana and Agustin Gargar. He finished his primary education at Malagasang II Elementary School, where he graduated in 2013. He took his secondary education at Kin Yang Academy, moved up on the year 2017 wherein he took Home Economics (HE) strand under the TVL track in his senior high school Kin Yang Academy and graduated on the year 2019. He is now currently taking a Computer Science course at Cavite State University – Imus Campus and is currently in his 4th year of college.

Ronneth Joy V. Gervacio was born on March 11, 2001, and is currently residing in Aniban 4, Bacoar, Cavite. She is the eldest child of Nenita and Raul Gervacio. She has a younger brother, John Ronette Gervacio. She finished her primary education at Aniban Central School, where she graduated in 2013. She took her secondary education at Bacoar National High School, graduated in 2017. She took Information and Communications Technology (ICT) strand under the TVL track in her senior high school and completed it in 2019. She is now taking a Computer Science course at Cavite State University – Imus Campus and is currently in her 4th year of college.

Khaizter Vash N. Topacio was born on January 14, 2001, and is currently residing in Poblacion, Imus, Cavite. He is the only child of Leo and Michelle Topacio. He finished his primary education at Imus Pilot Elementary School, where he graduated in 2013. He took his secondary education at General Emilio Aguinaldo National High School, moved up on the year 2017, wherein he took Information and Communications Technology (ICT) under the TVL track in his senior High School University of Perpetual Help System Dalta and graduated on the year 2019. He is now currently taking a Computer Science Course at Cavite State University-Imus Campus and is currently in his 4th year of college.

ACKNOWLEDGEMENT

The researchers would like to take this opportunity to express their sincere gratitude to the following individuals and organizations who have supported and assisted us throughout the process of completing this study.

First and foremost, the researchers would like to extend their deepest appreciation to their supervisors, **Mrs. Grace Ibañez** and **Mr. Klaid Bendio Moran**. Their guidance, expertise, and unwavering support have been invaluable in shaping this study. The researchers are grateful for their insightful feedback, encouragement, and dedication in helping them navigate through the complexities of this study.

The researchers are indebted to the staff and faculty of Cavite State University – Imus Campus, whose resources, facilities, and academic environment have provided an ideal platform for conducting this study. The department of Computer Studies and Information Technology has been instrumental in fostering a conducive atmosphere for intellectual growth and learning.

Furthermore, the researchers would like to acknowledge the assistance and support of their colleagues and fellow researchers. Their stimulating discussions, shared experiences, and camaraderie have greatly enriched our understanding and perspective.

The researchers are deeply grateful to the participants of this study, whose involvement and cooperation have made this study possible. Their willingness to share their insights and experiences has been invaluable in contributing to the knowledge base of this study. Although it is not possible to mention everyone who has contributed to this study, please accept their sincere appreciation for everyone's involvement and support.

GARGAR, GLENN ACE LIMWELL B.

GERVACIO, RONNETH JOY V.

TOPACIO, KHAIZTER VASH N.

Bachelor of Science in Computer Science

ABSTRACT

The goal of this research is to develop a game simulation that serves as an alternative learning tool for computer science and IT students, focusing on the basics of Python programming. The objective is to inspire self-study, encourage a creative learning approach, and familiarize students with problem solving work.

The research incorporates various modules to enhance the learning experience. The Lesson module consists of lectures and lessons that cover the Python programming topic, including tutorials, guided exercises, and challenges that require practical application of the concepts. This module seamlessly integrates with the gameplay, allowing students to actively engage with educational content while progressing through the game.

Iterative Development Cycle was used as the method in developing the game. Iterative development cycle includes Planning, Requirements, Analysis & Design, Implementation, Testing, and Evaluation before the desired deployment. By employing an iterative research design, this study aims to develop a Python programming simulation game iteratively, gathering user feedback and research insights along the way. This approach allows for continuous refinement,

By integrating these modules, the research aims to create an immersive and effective game simulation for alternative learning in basic Python programming. It offers an engaging and interactive environment that facilitates students' learning journey, enabling them to acquire and apply Python programming skills in a fun and educational way.

TABLE OF CONTENTS

SECTION	PAGE
BIOGRAPHICAL DATA	I
ACKNOWLEDGEMENT	II
ABSTRACT	III
CHAPTER I: INTRODUCTION	1
Statement of the Problem	3
Objective of the Study	4
Significance of the Study	4
Time and place of the Study	5
Scope and Limitation of the Study	5
Significance of the Study	6
Conceptual Framework of the Study.....	7
Definition of Terms	8
Participants of the Study	9
CHAPTER II: REVIEW OF RELATED LITERATURES AND STUDIES	10
Review of Related Literature	10
Review of Related Studies	15
CHAPTER III: METHODOLOGY	20
Research Design.....	20
Sampling Technique.....	23
Data to be Gathered.....	23
Statistical Treatment of Data	24

CHAPTER IV: RESULTS AND DISCUSSIONS.....	26
System Software Design	26
Discussion of Methodology Phases	40
System Software Evaluation Results	40
CHAPTER V: SUMMARY, CONCLUSION AND RECOMMENDATIONS.....	45
Summary.....	45
Conclusion	46
Recommendations.....	46
REFERENCES	48
APPENDICES.....	51

LIST OF TABLES

TABLES

1	Participants of the Study.....	9
2	Likert Scale	25
3	General Ratings of Participants in Functionality.....	40
4	General Ratings of Participants in Usability	41
5	General Ratings of Participants in Reliability	42
6	General Ratings of Participants in Efficiency	42
7	General Ratings of Participants in Portability	43
8	Overall Rating of the system.....	43

LIST OF FIGURES

1	Conceptual Framework of the Study.....	7
2	Iterative Development Cycle.....	21
3	Main Menu Screen	26
4	Credits Screen.....	26
5	Level Selection Screen.....	27
6	Tutorial Screen.....	27
7	Pause Menu Screen.....	28
8	Level 1 – Variables Level Layout.....	28
9	Interactable Dialogue Screen	29
10	Level 1 – Variables Topic	29
11	Level Compiler Simulation	30
12	“Phone” Simulation Screen.....	30
13	Diary Screen.....	31
14	Level 2 – Print/Input Level Layout.....	31
15	Level 2 – Print Topic.....	32
16	Level 2 – Input Topic	32
17	Level 3 – If/Logical Level Layout.....	33
18	Level 3 – If/Logical Topic.....	33
19	Level 4 – Arithmetic Level Layout	34
20	Level 4 – Arithmetic Topic	34
20	Level 5 – Loops Layout	35
20	Level 5 – Loops Topic	35

20	Level 6 – Lists Layout.....	36
20	Level 6 – Lists Topic.....	36
20	Level 7 – Dictionary Layout	37
20	Level 7 – Dictionary Topic	37
20	Level 8 – Function Layout.....	38
20	Level 8 – Function Topic	38
20	Game Final Assessment Screen	39
20	Game Assessment Score Screen.....	39

LIST OF APPENDICES

1	Time Table of Activities	52
2	System Evaluation Form	53
3	Curriculum Vitae.....	55
4	Sample Code.....	56

GAME SIMULATION FOR ALTERNATIVE LEARNING IN BASIC PYTHON FOR CS AND IT STUDENTS OF CAVITE STATE UNIVERSITY – IMUS CAMPUS

**GARGAR, GLENN ACE LIMWELL B.
GERVACIO, RONNETH JOY V.
TOPACIO, KHAIZTER VASH**

An undergraduate thesis manuscript submitted to the faculty of the Department of Computer Studies, Cavite State University Imus, Cavite in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science with Contribution No. _____. Prepared under the supervision of Ms. Grace S. Ibanez.

INTRODUCTION

Python is a strong, adaptable programming language that has grown in recognition over the past few years. Both novice and seasoned programmers favor it because of its straightforward and simple-to-read syntax. It has many different uses, including machine learning, artificial intelligence, and web development.

According to Smith, J. (2021), Python is an interpreted, interactive, object-oriented programming language. It provides high-level data structures such as list and associative arrays (called dictionaries), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc. It has a remarkably simple and elegant syntax and yet is a powerful and general-purpose programming language

Simulation, a type of technology by which theoretical knowledge is transformed into practice, simplifies the difficult-to-comprehend, abstract activities and realistically embodies them and can reach a solution in a short time (Atalan and Donmez, 2019).

In recent years, digital or web-based games have increasingly supported learning. In the context of online education, this research area attracts a significant amount of interest from the scientific and educational community, for example tutors, students and game designers.

With the growing expansion of technology, instructors and those who create educational policy are interested in introducing innovative technological tools, such as videogames, virtual worlds, and Massive Multi-Player Online Games (MMPOGs) (Buckless, 2014; Gómez, 2014).

According to Gagne (2005) It is well documented in the literature that intrinsic motivation in learning activities is strongly correlated with the outcome of the learning process, by promoting direct participation of students in their knowledge building.

Moreover, real-life systems are usually subject to uncertainty and dynamism, some of which are generated by the human component of the system (Gruler et al. 2019). These aspects cannot be easily included in traditional analytical models, and typically require the use of simulation-based methodologies in coordination with other techniques, such as heuristic-based optimization and machine learning methods. As pointed out by Juan et al. (2017), the use of simulation software, tools, and games facilitates the practical understanding of these complex systems and allows students to enhance their learning experience via the development of hands-on activities properly designed by their instructors.

According to Dimitrios Vlachopoulos & Agoritsa Makri (2017), Games and simulations show mixed effects across a number of sectors, such as student performance, engagement, and learning motivation. However, as these studies focus only on certain disciplines, there remains a gap in the literature concerning a clear framework of use across academic programs. As a result, the issue of efficiently integrating games and simulations in the educational process is often up to the instructor's discretion.

Accordingly, this research seeks to offer a thorough understanding of the Python programming language, as well as its potential as a potent tool for creating a wide variety of applications. This will aid students in developing their logic and problem-solving skills, as well as their understanding of basic Python Programming Language fundamentals. This teaching strategy imparts knowledge to students by utilizing advantageous elements of video games.

Statement of the Problem

The issue with developing a game as a teaching tool for Python programming is that it needs a lot of development resources and know-how. It might be difficult to design a game that effectively teaches programming fundamentals and is also entertaining to play. The game must also be adaptable to players of varied programming skills, from absolute beginners to experts. To make sure the game is both informative and interesting, a lot of preparation and design is needed. Another issue is that the game should be able to run on several platforms and have an intuitive user interface with comprehensive training. The game should also have a feature that allows users to track their progress and receive performance-related feedback.

The game should also be flexible enough to accommodate various learning styles by offering a variety of ways to engage with the programming principles, including visual aids, interactive simulations, and quizzes. By offering a method to monitor progress and rewarding players for performing specific objectives, the game should be able to provide users a sense of satisfaction.

Specifically, this research aims to answer the following specific problems:

Having a poor foundation in Python Programming Language can lead to students under the program of CS and IT, having a hard time understanding complex theories, like general syntax, basic structures such as functions, data structures, and modules. *“How can learning Python provides basic learning experience through game?”*

A lot of CS and IT students are having a hard time in familiarizing Python Coding. A lot of times, they lack in practice, reading modules, and studying open-source code projects. *“How to develop a system that provides learning in an easy way?”*

At first, some of CS and IT students, when it comes to coding, they are all trying to memorize the codes. They try to memorize the codes instead of understanding the syntax. *“How to provide an alternative way to test basic python codes?”*

Objective of the Study

The goal of this research is to create a game that will act as a teaching tool for a computer science course regarding basic python programming language. The goal of this research is to inspire everyone to engage in self-study, pursue the subject with a creative learning approach, and get familiar with scientific work.

Specifically, this research aims to answer the following questions:

1. To provide a game that serves as a teaching tool for Python Programming Language.
2. To develop creativity and ingenuity in task completion and problem resolution.
3. To create a proper step-by-step understanding in algorithm provided for the users.

Significance of the Study

The developed system which is entitled as “Game Simulation for Alternative Learning in Basic Python”, is aimed to assist students in learning, with or without prior knowledge regarding Python Programming Language.

Since games may support a number of learning-promoting variables, including student motivation, active learning, capacity to adapt, cooperation, and simulation, they could serve as an excellent tool for enhancing teaching the said subject. The developed system is beneficial to the following entities;

CS and IT students. This research will help to fast track information that they already learned and relevant analytics that will help in constructing robust programs. The developed system can serve as a pick-me-up video game or a refresher video game for students who are taking Computer Science (CS) and Information Technology (IT) courses.

Faculty. This research will be beneficial to every faculty member, such as teachers and professors. The developed system will aid them in educating the students in view of the fact that the proposed system allows them to practice coding and learn the Python Programming in a more engage way.

Without prior knowledge to Python Programming Language. This research will be beneficial to people without prior knowledge to Python Programming Language as it creates opportunities, and ensures all information are timely for that matter. Such as people who are interested in learning Python Programming Language but has a limited knowledge.

Future Researchers. This research will help them in innovating, developing, and modifying this related study.

Time and Place of the Study

The study entitled Game Simulation for Alternative Learning in Basic Python for CS and IT Students of Cavite State University – Imus Campus, which began in March 2023 and was approved for Title Defense on March 2023 and will be finished in July 2023.

The researchers had a weekly consultation with the Thesis Adviser and Technical Adviser regarding the application and documentation and conducted interviews at the prospective campus of Cavite State University in March to May 2023.

The researchers had an in-person System Presentation and Evaluation in July 2023

Scope and Limitation of the Study

The purpose of this study is to develop and execute a Game Simulation for Alternative Learning in Basic Python Programming. Furthermore, the study concentrated on the following

Topic module. The game has a collection of lectures or lessons that cover the topic of Basic Python programming. It includes a topic that is written in text about certain information that require the player to apply their learning. The module is typically integrated seamlessly into the gameplay experience, ensuring that the player can actively engage with the educational content while progressing through the game.

Tutorial module. The developed application aims to provide a structured and interactive learning experience for students to acquire Python programming skills within the context of game-based simulations, where it includes instruction on how the user will be able

to run the application. The tutorial module aims to enhance students' understanding, application, and retention of Python programming skills, thereby contributing to their overall learning outcomes.

Assessment module. The developed application has an assessment after every lesson. In essence, the assessment serves as a critical component of the simulator, as it helps establish the credibility of the system and also provides a clear understanding of the said topic. Formative assessments will be integrated throughout the game-based learning experience to provide ongoing feedback and monitor students' progress. These assessments can take the form of in-game quizzes, coding challenges, or interactive exercises that allow students to apply their knowledge and skills in real-time.

Simulation module. The developed application designed to mimic activities you'd see in the real world such as Python Programming Language, Compiler and Assessments. The purpose of the game may be to teach you something.

Compiler module. The developed application has a software component that converts the code that can be executed by the simulator. It is responsible for analyzing the code, checking for syntax errors, and translating the code into a form that can be understood by the simulator.

Level module. This aims to provide a balanced and engaging gameplay experience that fosters continuous learning of Basic Python programming concepts, and the development of problem-solving skills. The developed game focuses on the design and progression of levels within the game-based simulations, this includes Level Design, Learning Objectives and Progressive learning. The level module have levels or puzzles that the user can access after pressing the play button on the boot up of the game or in the main menu.

There could be some limitations in regards the research. The limited scope of the research, which may not cover all aspects of using videogames as an educational tool or the Python programming language.

The system may not be able to account for all individual differences in learning styles and preferences, which can affect the effectiveness of the game as a learning tool.

The system may not be able to account for all factors that can influence the effectiveness of the game as a learning tool, such as the prior knowledge and experience of the players.

The limited resources and time, which can affect the development and testing of the game.

Conceptual Framework of the Study

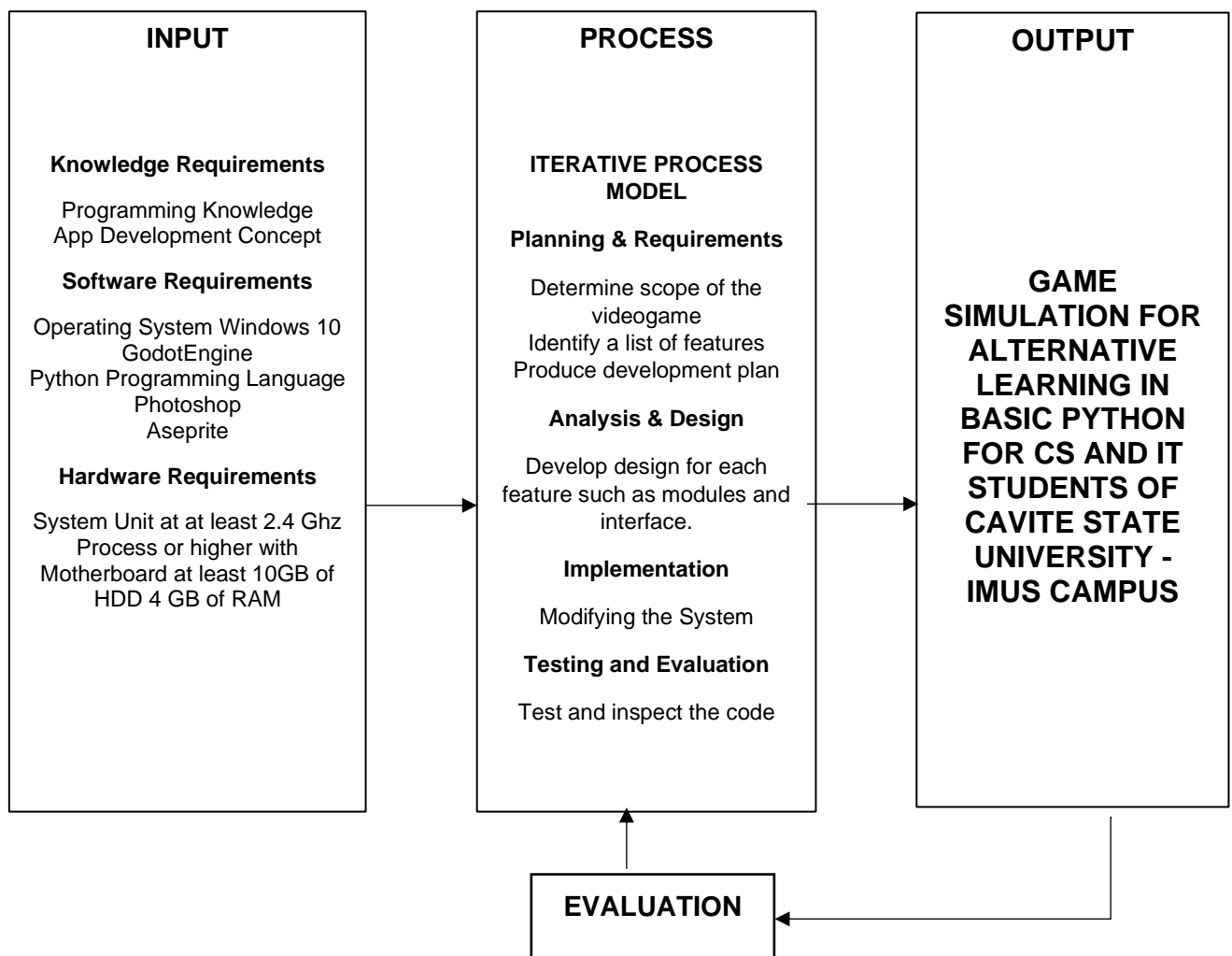


Figure 1. The conceptual framework of the study

The proponents used the Input-Process-Output model for the conceptual framework of the study. Input phase lists the requirements needed in conceiving the study – from knowledge, software, up to hardware area. The Process shows the phases of software development model the proponents will undergo in building the videogame. Lastly is the Output which is the culmination of the Input and the Process phases, leading to the development of Python Programming simulation through video games for CS and IT students of Cavite State University.

Definition of Terms

Videogame. A videogame is an electronic game that can be played on a device. The game uses visual and audio feedback to interact with the player, and typically involves controlling a character or object in a virtual environment to complete tasks or achieve goals. Video games can be single or multiplayer, and can be categorized into various genres such as action, adventure, sports, and puzzle.

Python (Programming Language). Python is a high-level, general-purpose programming language that is widely used for web development, scientific computing, data analysis, artificial intelligence, and more.

Game Engine. A game engine is a software development environment that is specifically designed to create video games. It provides a set of tools and frameworks for game developers to create and run the game

Development. Development refers to the process of creating, designing, and implementing a product or system. This process can involve various stages, such as planning, research, prototyping, testing, and deployment.

RPG. A game in which participants take on the roles of characters in a fictional scenario. Players are responsible for physically carrying out these roles within a narrative, either through literal acting or through a structured decision-making process about character development.

NPC (Non-Playable Character). A character in a computer game that is not controlled by someone playing the game, mostly interactable by the player.

Participants of the Study

The researchers selected a number of CS Students, IT Students, and other respondents as participants in the development of the application. The selection of participants for the study is a critical process that ensures the research finds the target audience, the researchers chose Students and other respondents as respondents to become players on the videogame as they would test the system. After that, they would provide feedback about the system and let the researchers know if the developed system could pass the strict and standard requirements of the professionals. And later on the research the researchers would try to evaluate and modify the said videogame.

Table 1: Classification of the respondents of the application

Classification	Frequency	Percentage
CS Students	26	52%
IT Students	21	42%
Other Respondents	3	6%
Total	50	100%

REVIEW OF RELATED LITERATURES AND STUDIES

In this chapter, the researchers' most recent research is discussed in relation to relevant prior literature and studies.

Related Literature

Game-play Experience

Schell, Jesse, Game-play experience is the last model in my finite definition of game design. Included in my definition are game mechanics and play mechanics, which represent a means to an end

The game-play experience model begins with questioning how a person has an experience. Effects of gaming on players have been explained by cognitive mechanisms, semiotic approaches and compared to other forms of media.

Gaming triggers the use of a player's basic faculties and challenges them to improve these faculties. The game-play experience model claims that games not only require the use of your faculties to play them, but gaming is an activity directly aimed at triggering their use, challenging them and ultimately encouraging their improvement.

The Many Faces of Role-Playing Games

Role-playing games have evolved into many forms in their thirty-year history. From the traditional pen-and-paper form, that originated with Dungeons & Dragons, with a group of friends playing around a table, to large live-action role-playing game, with hundreds of people acting out their assumed roles.

The first computer role-playing games appeared over twenty-five years ago and massively multi-player role-playing games, such as World of Warcraft are now one of the most popular genres of digital games.

Hampering any attempt to understand what makes a game a role-playing game is the subtle divide between role-playing and role-playing game. Role-playing can take in many places, not all of them games (such as ritual, social activities, therapy, etc). This means that definitions of the role-playing activity are not that useful in separating role-playing games from other games. In this paper the researchers started from the position that the players are correct: they know what a role-playing game is. By examining a range of role-playing games some common features of them emerge. This results in a definition that is more successful than previous ones at identifying both what is, and what is not, a role-playing game.

Game-based Learning and 21st Century Skills

Game-based learning and 21st century skills have been gaining an enormous amount of attention from researchers and practitioners. Given numerous studies support the positive effects of games on learning, a growing number of researchers are committed to developing educational games to promote students' 21st century skill development in schools. However, little is known regarding how games may influence student acquisition of 21st century skills. This paper examines the most recent literature in regard to game-based learning and identified 29 studies which targeted 21st century skills as outcomes. The range of game genres and game design elements as well as learning theories used in these studies are discussed, together with the range of indicators, measures and outcomes for impacts on 21st century skills. The findings suggest that a game-based learning approach might be effective in facilitating students' 21st century skill development. The paper also provides valuable insights for researchers, game designers, and educators in issues related to educational game design and implementation in general.

Game design and play require people to be familiar with media and technology, and it also requires people to be creative and critical thinkers, so it has great potential to facilitate students' 21st century skill development. Given the lack of consistent empirical evidence with respect to the effectiveness of game-based learning, this review aims to examine the most

recent literature regarding game-based learning and seeks to further understand the influence of games on learning, with a major focus on students' 21st century skill development.

The Effect of Games and Simulations on Higher Education: A Systematic Literature Review

The focus of higher education institutions is the preparation of future professionals. To achieve this aim, innovative teaching methods are often deployed, including games and simulations, which form the subject of this paper. As the field of digital games and simulations is ever maturing, this paper attempts to systematically review the literature relevant to games and simulation pedagogy in higher education. Two researchers collaborate to apply a qualitative method, coding and synthesizing the results using multiple criteria. The main objective is to study the impact of games and simulations with regard to achieving specific learning objectives. On balance, results indicate that games and/or simulations have a positive impact on learning goals. The researchers identify three learning outcomes when integrating games into the learning process: cognitive, behavioral, and affective. As a final step, the authors consolidate evidence for the benefit of academics and practitioners in higher education interested in the efficient use of games and simulations for pedagogical purposes. Such evidence also provides potential options and pathways for future research.

The researchers used a lot of research methods. The authors developed a pre-defined review protocol to answer the research questions, specifically aimed at minimizing researcher bias. The literature review was carried out between July and October 2016 and followed the design stages described below.

The reviewed papers are identified through keywords in referenced electronic databases, such as Google Scholar, Web of Science, ERIC, PsycInfo, PsycArticles Fulltext Search, InterDok, ProQuest, Scopus, BEI, and SearchPlus. The keywords for learning outcomes are a combination of the term games or simulations paired with the term higher education, employing the Boolean operator "AND". Additional keywords for learning outcomes are learning objectives, learning goals, learning objectives and effects. Keywords for platform

and delivery methods include computer-based, web-based, digital, virtual, online, and technology. Keywords for games and simulations are educational games, business simulations, role-playing simulations, game-based learning, video games, and serious games. Moreover, the Boolean operator “OR” is employed to combine all these keywords. The study sets the broadest range of keywords, so as not to limit the scope of related articles.

Furthermore, the researchers conducted a comprehensive database search in bibliographic indices for the data selection. The search is related to a variety of scientific fields of study, including Education, Psychology, Information Technology, Management, and other scientific areas (e.g., Engineering, STEM, Health, etc).

Python Programming Language

Python is a computer programming language that is frequently used to create websites and applications, automate operations, and analyze data. Python is a general-purpose programming language, which means it can be used to develop a wide range of applications and is not specialized for any particular problem.

In accordance with Stack Overflow's 2022 Developer Survey, Python is the fourth most popular programming language, with respondents stating that they use Python about 50% of the time in their development work. According to survey results, Python is tied with Rust as the most desired technology, with 18% of developers who aren't already using it expressing an interest in learning Python.

Purpose of Python

In line with the article published on Future Learn, Python was first introduced in 1992 and was designed in such a way that it's relatively simple to write and understand. Some of the organizations that utilize it are from NASA. Google, Spotify, and a plethora of others use the language to power their services. (Fran, 2021)

Needless to say, Python is one of the most popular programming languages in the world, contributing from what we can see now in Netflix's recommendation algorithm to the software that controls self-driving cars. Python is classified as a general-purpose language, this implies it's intended for usage in a variety of applications, such as data research, software and web development, automation, and general task completion.

There are a lot of things Python can offer once you learn from it, and in this study, we would like to focus on making a Python interactive video games that can help students in the program of Computer Science (CS) and Information Technology (IT).

Python for Beginners

The article Reak Python considers that the first step in learning any programming language is to understand how to learn. What is the value of knowing how to learn? The answer is easy to grasp: as languages evolve, libraries and tools are developed. Keeping up with these changes and becoming a successful programmer will require knowing how to learn. (Hudson-Hurley, 2023)

Python's basic data structures include list, set, tuples, and dictionary. Each data structure is distinct in its own way. Data structures are "containers" that organize and categorize data. (Taylor, 2020)

Backtracking Algorithm

Backtracking is a general algorithm for solving some computational problems, most notably constraint satisfaction problems, that incrementally builds candidates to the solutions and abandons some candidates backtracks as soon as it determines that the candidate cannot be completed to a reasonable solution. (Upadhyay, 2023)

It finds a solution by building a solution step by step, increasing levels over time, using recursive calling. A search tree known as the state-space tree is used to find these solutions. Each branch in a state-space tree represents a variable, and each level represents a solution.

A backtracking algorithm uses the depth-first search method. When the algorithm begins to explore the solutions, the abounding function is applied so that the algorithm can determine whether the proposed solution satisfies the constraints. If it does, it will keep looking. If it does not, the branch is removed, and the algorithm returns to the previous level.

Graphic Design in the Video Gaming Industry

Video gaming is one of the most popular forms of entertainment in the world. It is a billion-dollar industry, and it is only getting bigger. Graphic design is a critical part of the video gaming industry. It is responsible for creating the look and feel of the games. (Boodie, 2023)

Good graphic design can make a game look and feel immersive and realistic. It can also make a game more fun and engaging to play. Unfortunately, many video games have poor graphic design. This can make the games look ugly and unappealing. It can also make them difficult to play.

Related Studies

Game-based approaches for specializing in Information Technology

Game-based learning and gamification have become popular terms in teaching methodology in education of late. They are both generating considerable interest in terms of complementing traditional teaching techniques, providing diversity in teaching methods, enlivening teaching topics. Games are also good for students as they increase their motivation, develop creative thinking and allow practicing the language in a fun and relaxed way, incorporating challenges into learning process, focusing on the task given and avoiding the failure.

In teaching English as a foreign language, games were used as ice-breakers, warmups, in the end of the lesson if there is extra time or during the lesson to add some

variety. Researcher highlights that “many experienced textbooks and methodology manuals writers have argued that games are not just time-filling activities but have a great educational value. W. R. Lee holds that most language games make learners use the language instead of thinking about learning the correct forms. He also says that games should be treated as central not peripheral to the foreign language teaching programmer. A similar opinion is expressed by Richard-Amato, who believes games to be fun but warns against overlooking their pedagogical value, particularly in foreign language teaching (Hadfield, 1999a).

Moving from Waterfall to Iterative Development: An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP

The Rational Unified Process (RUP) is a software process framework developed by Rational Software, from earlier work in the 1990s. Its main aim was overcoming several limitations of the software process practices at that time (specifically the “classic” waterfall model). RUP was designed by combining the most common best practices in the 90s and resulted in an iterative, incremental software process which is risk-driven and architecture-centric.

This study describes an exploratory case study of the benefits of using an iterative development process versus a waterfall process. The study was performed in a division of an IT department of a large multinational company. A set of people in different roles in this organization were interviewed. Researchers triangulated the results with the opinion of experts from other companies and existing literature. The research finds several categories of factors for which the use of iterative development is preferable over waterfall development in terms of business value. Based on the study, as it is being reviewed and discussed, it's possible to isolate flaws in functions or design. Finding these issues at an early stage may help to address them quickly within a tight budget. It enables them to swiftly implement changes and then get customer feedback to guide product decisions. Because the modifications are not drastic, they are simpler to manage and test.

Godot Engine and Checklist-Based Specifications: Revising a Game Programming Class for Asynchronous Online Teaching

This study describes the revisions to an undergraduate elective game programming course that were made in response to the COVID-19 pandemic. The course transitioned from a lab-based, in person class to an online, asynchronous one. This required a change in teaching technology from Unreal Engine 4 to Godot Engine. The course expanded its use of checklist-based specifications grading in order to facilitate student autonomy with minimal reduction in creativity and motivation. The course revisions required significant time investment, but the results were positive.

A game engine is a software system that combines several tools and libraries that are useful for game development. These are often brought together into one interface in the same way that an integrated development environment provides for general software development. Unity, Unreal Engine, and GameMaker are three popular game engines.

Godot Engine is a game engine with several noteworthy features, including: supporting both 2D and 3D game development; running on multiple platforms; building for multiple platforms, including HTML5 and mobile; tooling for 2D and 3D animation; and a custom scripting language—GDScript—that is heavily inspired by Python.

Godot Engine has significantly more modest hardware requirements than UE4, satisfying this important constraint. Additionally, the project website at godotengine.org includes extensive documentation specifically to help beginners. Godot Engine is free and open source software (FOSS), licensed under the Expat (“MIT”) license.

CS circles: an in-browser python course for beginners

Computer Science Circles is a free programming website for beginners that is designed to be fun, easy to use, and accessible to the broadest possible audience. This website teaches Python since it is simple yet powerful, and the course content is well-

structured but written in plain language. The website has over one hundred exercises in thirty lesson pages, plus special features to help teachers support their students. It is available in both English and French. We discuss the philosophy behind the course and its design, we describe how it was implemented, and we give statistics on its use.

Animated vs. Static graphics in a video game

In this study, two separate versions of the game were developed: one version which included a higher level of animation, and one where the graphics were predominantly static, including only necessary animations. Researchers ensured that the game mechanics were not affected by the animations in any way and that the different versions would be as similar as possible, discounting the animations. Some animations that were regarded to be connected to interactivity were left in the game, even in the static version. This was to be certain that the study would evaluate the effect of animations, and not the interactivity.

The general impression that the data left the researchers with was that the two versions were not perceived as very different from one another. Thus the animation or lack thereof had little to no impact on the user experience in the tests conducted, at least when it comes to positive/negative affect and immersion.

Research on the Application of Pixel Art in Game Character Design

This paper analyzes the development, basic design foundations, and design principles of character designs that are often found in today's games. This discussion relates to the ability of pixel styles to survive even though the design styles of the current digital art era are developing with diverse thoughts. Pixels are the basic elements of design that are capable of producing attractive design work. Looking at the current development of pixel styles, in the midst of the proliferation of 3D-based visual works, we can find that pixel styles still have an irreplaceable influence. Pixel-based designs are still being put into practice and applied in the form of games by more designers and artists

While it is undeniable that 3D images have become the design mainstream, the pixel art style still has an indelible influence. The pixelated design, as an exploration between retro and trend, also attracts the hearts of many new consumers with its uniqueness. In design, the pixel style is independent and permeates all aspects of people's lives. An unit of blocks are put together to abstractly show various forms, but they also have a unique beauty.

Pixel art is a visual approach that is able to produce visual signs that can distinguish one character from another. This technique is also a technique that allows the realization of a game with a production process that is simpler and cheaper This advantage can support game artists to produce games need media to more productively or even educators who need media to enrich the teaching process which is often referred to as edugame.

RESEARCH METHODOLOGY

This chapter covers the research methodology done by the proponents – from research designing to data gathering method and how it will be treated then applied in building the system. It will also discuss the specifications of materials, equipment, and other requirements necessary for the project. The minimum specification of device required in running the system smoothly will also be discussed. Mixed research methods would be used to gather numerical data that can be analyzed and quantified, such as player performance, player engagement, and player satisfaction. These methods would include user testing, surveys, and data analysis.

Research Design

In developing the system, the researchers employed several models and approaches to outline the project and to precisely define its needs. An iterative methodology was used in developing the game to be able to come up with a videogame that was made systematically. Godot Game Engine were used in encoding and editing, Github for code hosting tools for repositories management and collaboration, and Aseprite and Photoshop were used to modify and create art assets that is used in the videogame.

The researchers used the Iterative Development methodology in developing the system. Iterative development involves developing the project in multiple iterations or cycles, with each iteration building upon the previous one. In the context of game development for research purposes, an iterative approach allows for continuous refinement and improvement of the Python programming simulation game based on user feedback and research insights. Iterative development includes steps as shown on Figure 1. These are Planning; Requirements; Analysis & Design; Implementation; Testing; Evaluation; and Deployment.

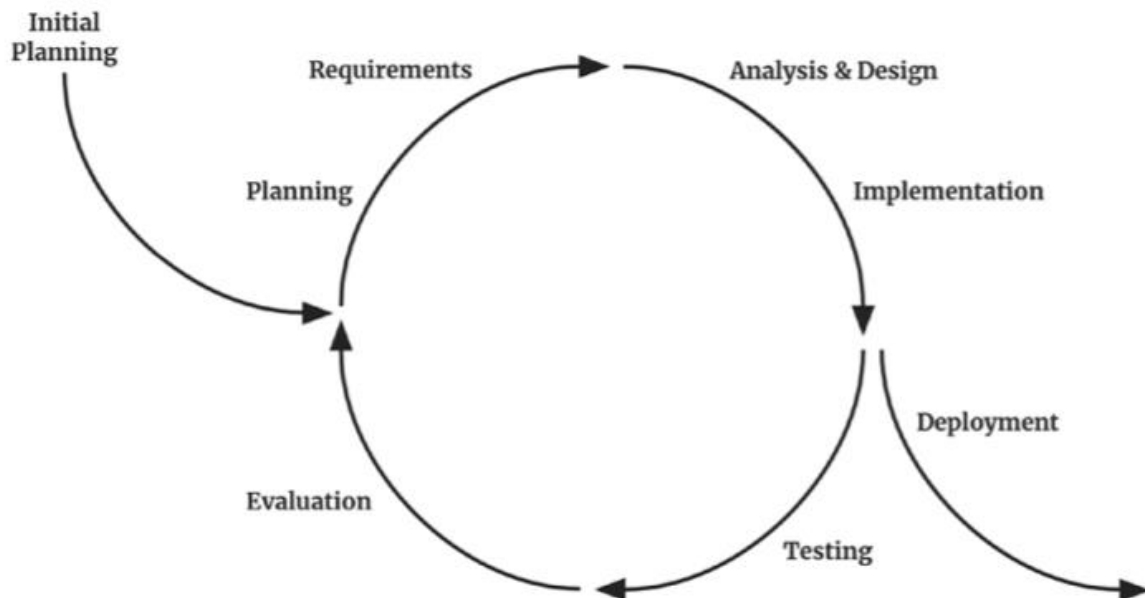


Figure 2. Iterative Development methodology

The researchers used Iterative Development because it suits the videogame development. By employing an iterative research design, this study aims to develop a Python programming simulation game iteratively, gathering user feedback and research insights along the way. This approach allows for continuous refinement, ensuring that the game effectively supports learning and provides an engaging educational experience. The results obtained will contribute to the field of programming education and provide valuable insights into the use of video games as a learning tool for Python programming.

Planning & Requirements Phase

During this phase, the main goal of the researchers is to gather the necessary modules for the system. To achieve this, the researchers distributed questionnaires to potential users and gather relevant information and references about the problem at hand using books that have related topics and research papers.

Analysis Design & Implementation Phase

When the necessary information is gathered after the determination of the requirements phase is complete, the researchers will identify the design to implement and potential risks and difficulties that the system may encounter. To proceed, the researchers will evaluate the possible implications of these risks and devise methods to either fix them or reduce the risk impact

Development and Testing Phase

During the first few rounds of system development, the developers' main goal is to concentrate on creating the system's fundamental elements. These are the modules that will serve as the framework or core of the system. Core modules make an effort to address the issue even in the early stages of development. Then the developers will perform testing, which involves assessing the system's capabilities to find issues or errors that could affect the data that the system will handle. Following testing, a limited sample of potential users will utilize and test the system. The developers will then collect their input and feedback to utilize it to improve the system.

Evaluation and planning the next Iteration Phase

After going through the testing and development processes, the system is finally prepared for release. Later, the general public will be able to use the system. When utilizing the system, users could have bugs and problems, which will be fixed by the developers. Following version stabilization, the creators will gather further pertinent feedback and look for opportunities to increase the system's functionality. The researchers will then plan the next modules to be added to the system, and the cycle will repeat until the system is finished.

Research Locale

The study was entitled Game Simulation for Alternative Learning in Basic Python which began in March 2023 and will be presented on July 2023. Conducted in Cavite State University – Imus Campus, which is located at Land Transportation Office (LTO) Compound at Emilio Aguinaldo Highway, Imus City, Cavite.

Sampling Technique

The researchers used convenient sampling in conducting the research. Convenience sampling is a qualitative research sampling strategy that involves selecting participants based on their accessibility and availability to the researcher. This sampling technique is cost-effective and time-efficient method of data collection

The researchers choose the members of population to participate in the study – which are people that are CS Students, IT Students, and Other Respondents.

Data to be Gathered

The researchers conducted surveys and questionnaires as a mean of gathering and collecting data. It is a formal survey form used as a means of obtaining data from targeted respondents with the goal of generalizing the results to a broader public. The surveys and questionnaires were all about what problems they encountered on Python Programming Language, and everything about they think regarding Python Programming Language. The proponents also gave questions about the present system they used alongside with the problems they used to encounter when using it. There was also a question about the features they wanted to have or be changed for the possible future that they can use in which will result to an effective and efficient environment for their concerns.

Statistical Treatment of Data

The response will be analyzed using descriptive statistics statistical tools. To ascertain the precise interpretation of the results, the material acquired was tallied and processed manually and with a computer. Data was gathered, tabulated, and analyzed.

The following statistical tools were used in the presentation and analysis of data gathering in the study.

Mean. This is the calculated “central” value set of a set of the numbers

$$\bar{X} = (\sum x) / N$$

Where:

$$\bar{X} = \text{Weighted Mean} = (fd * 1) + (fd * 2) + (fd * 3) + (fd * 4) + (fd * 5) / \text{Total Number of Population}$$

x = weight of each proportion

\sum = summation

N = total number of population.

fd = frequency distribution

Frequency and percentage. This measures the weight of frequency of respondent's data. The formula is: $P = (f / n) * 100$

Where:

P = percentage

f = frequency

n = number of respondents

Likert scale. Likert's scale was used to interpret the survey forms. The users were asked to evaluate the application in terms of its functionality, efficiency, portability, usability and reliability.

Table 2: Likert Scale

Rating	Scale Interval	Vertical Interpretation
5	4.50 - 5.00	Excellent
4	3.50 - 4.49	Very Good
3	2.50 - 3.49	Good
2	1.50 - 2.49	Fair
1	1.00 - 1.49	Poor

Table 2 shows the interpretation of the ranges probability for evaluating the scores on the evaluation and the ratings of each interpretation to determine the results of the evaluation.

RESULTS AND DISCUSSION

This chapter covers the results, interpretation and analysis of data gathered that are presented in figures, textual, and tabular form with the aid of statistical treatment of analysis and interpretation purposes.

System Software Design



Figure 3: Main Menu Screen

Figure 3 shows the first screen all of the users will see. This screen will allow users to choose whether they want to play, reset data or exit

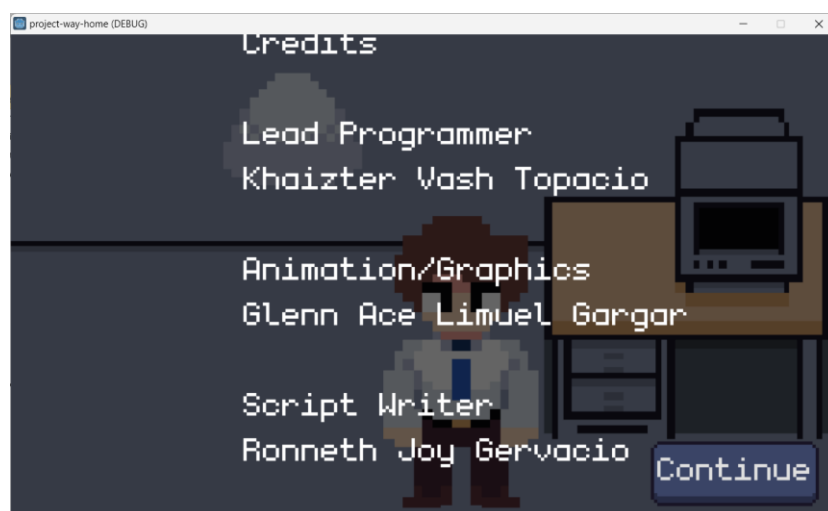


Figure 4: Credits Screen

Figure 4 shows the Credit screen when the user selects the Credits button, where users can see the people that made the application happen.



Figure 5: Level Selection Screen

Figure 5 shows the level selection screen when the user selected the play button, where users can select which level to play on. Levels are unlocked based on what are already completed by the user that is saved locally.



Figure 6: Tutorial Screen

Figure 6 shows the Tutorial Screen. This screen is where users can see how to control the player movement and how to interact to certain objects. Users can access this tutorial on the diary section or on the pause button.



Figure 7: Pause Menu Screen

Figure 7 shows the Pause Menu Screen. Users can access this screen by pressing the pause button on the top right corner of the game. Users can select the buttons to continue, go back to the Main Menu Screen or Exit the Game entirely.



Figure 8: Level 1 – Variables Layout

Figure 8 shows the screen where the user can see the layout of the Level 1 – Variables.



Figure 9: Interactable Dialogue Screen

Figure 9 shows the Dialogue Screen. The dialogue screen will popup when interacting with an interactable object.

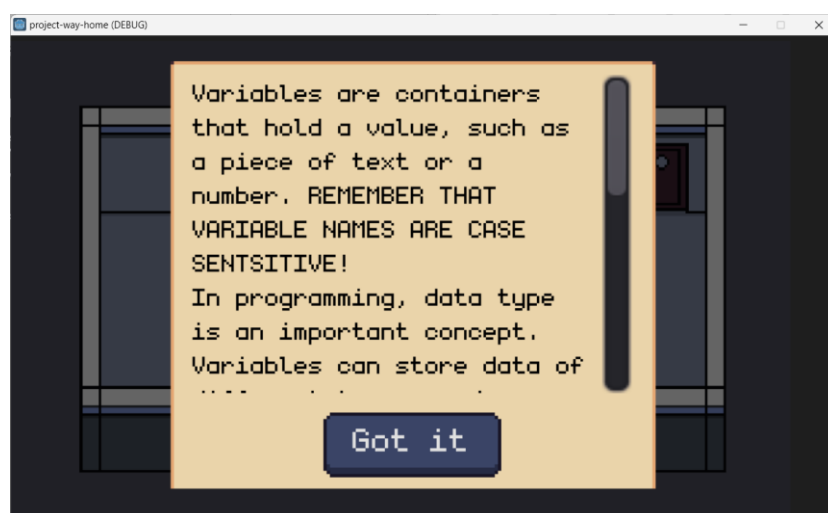


Figure 10: Level 1 – Variables Topic

Figure 10 shows the screen for Level 1 – Variables Lesson. It shows what the assessment on the level could be and shows lessons about the level.

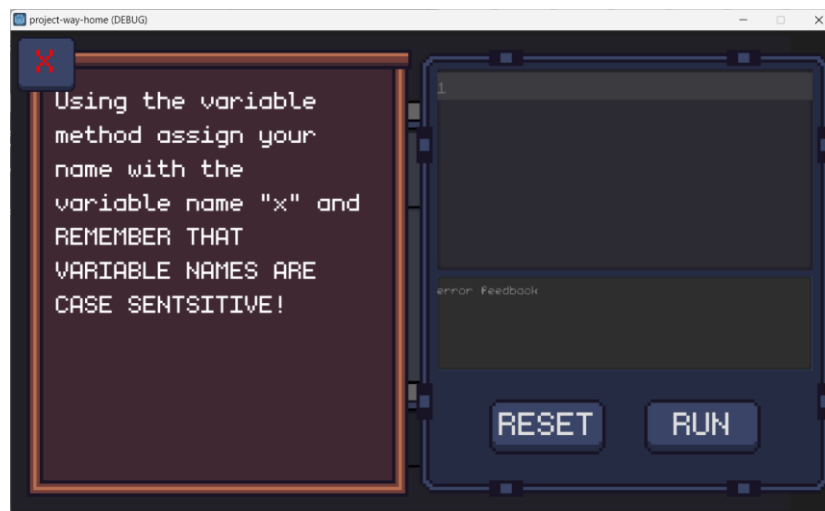


Figure 11: Level Compiler Simulation

Figure 11 shows the Level Compiler Simulation. A Simulation of a python programming compiler will pop-up after interacting with the level assessment interactable object and must input the given assessment using the lesson learned on the level in order to proceed.

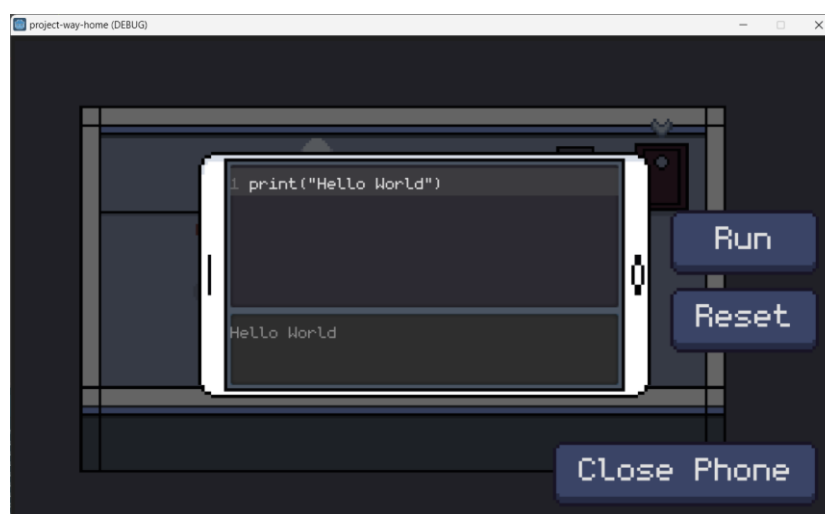


Figure 12: "Phone" Simulation Screen

Figure 12 shows the "Phone" Screen. It is a Simulation of a compiler that users can use freely and not restricted by the level assessment. The Run button runs the code, the reset button resets any input on the "Phone", and the Close Phone button closes the "Phone" Simulation Screen

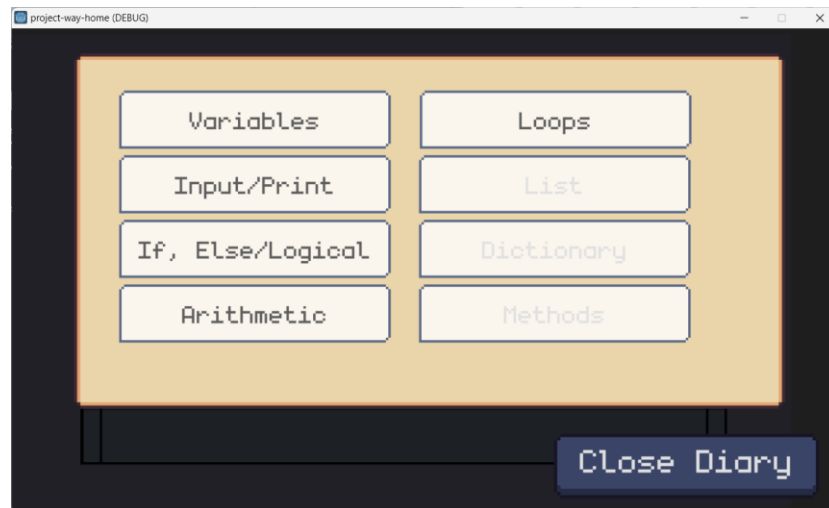


Figure 13: Diary Screen

Figure 13 shows the Diary Screen. The user can check the diary to relearn past lessons from the past levels.



Figure 14: Level 2 – Print/Input Layout

Figure 14 shows the Level 2 – Print/Input Layout. This is the layout of the level 2 section.

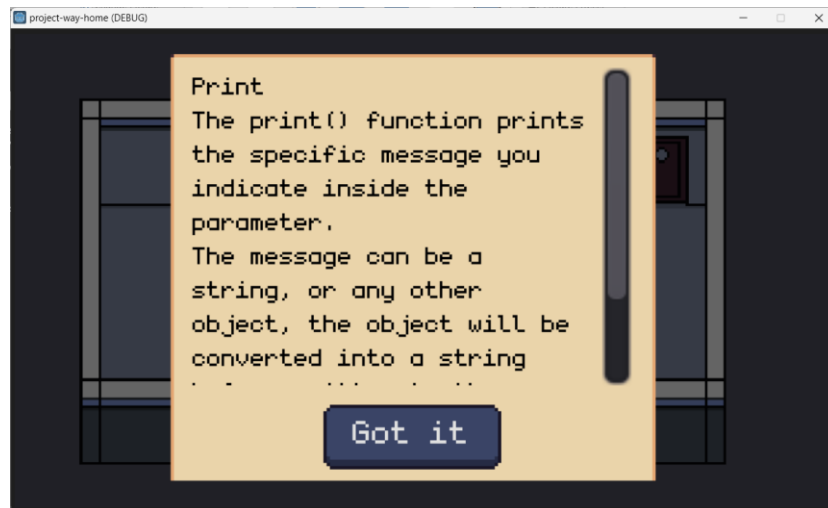


Figure 15: Level 2 – Print Topic

Figure 15 shows the Level 2 – Print Lesson. The user gets prompts about the lesson on lesson 2 about printing.

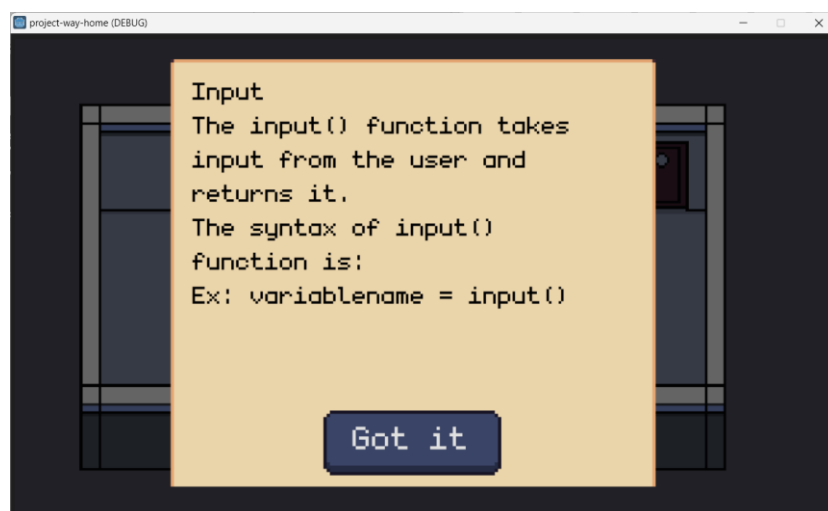


Figure 16: Level 2 – Input Topic

Figure 16 shows the Level 2 – Input Lesson. The user gets prompts about the lesson on level 2 about input.



Figure 17: Level 3 – If/Logical Layout

Figure 17 shows the Level 3 – If/Else/Logical Layout. The user sees this layout of the level upon entering level 3 of the game.

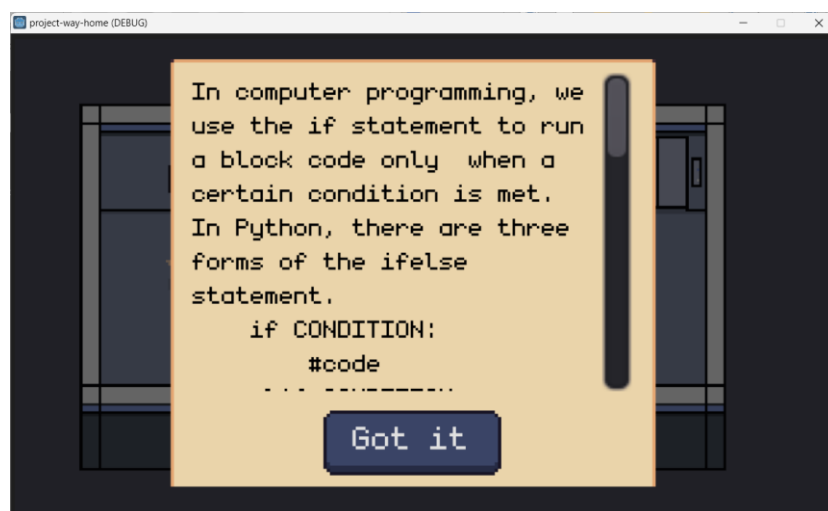


Figure 18: Level 3 – If/Logical Topic

Figure 18 shows the Level 3 – If Lesson. The user gets prompts about the lesson on level 3 about if statements with the logical operators “OR” and “AND”.

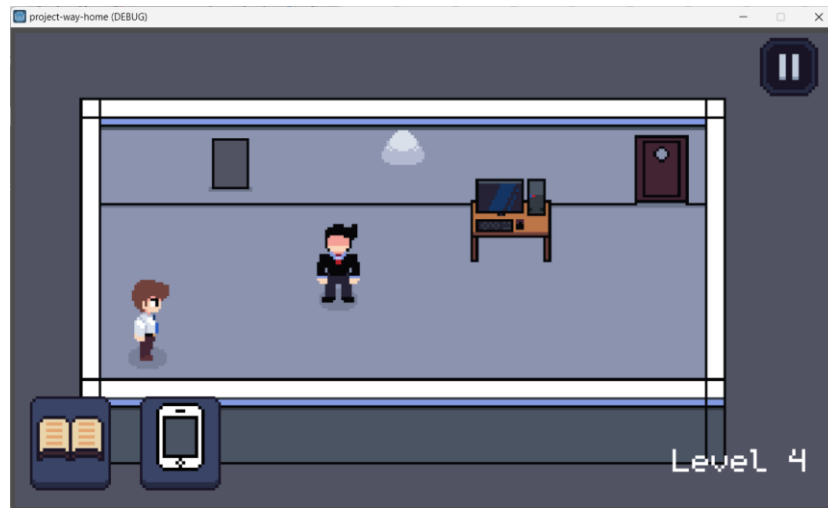


Figure 19: Level 4 – Arithmetic Layout

Figure 19 shows the Level 4 – Arithmetic Layout. The user sees this layout of the level upon entering level 4 of the game.

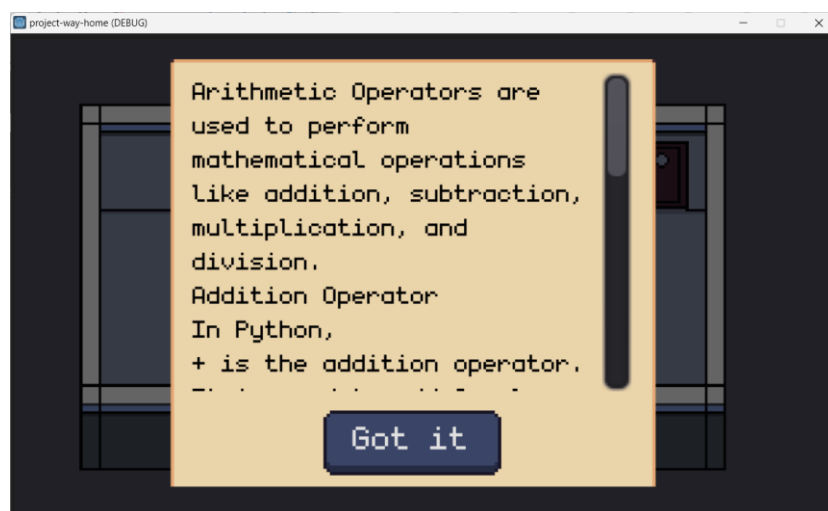


Figure 20: Level 4 – Arithmetic Topic

Figure 20 shows the Level 4 – Arithmetic Topic. The user gets prompts about the topic on level 4 about arithmetic.



Figure 21: Level 5 – Loops Layout

Figure 21 shows the Level 5 – Loops Layout. The user sees this layout of the level upon entering level 5 of the game.

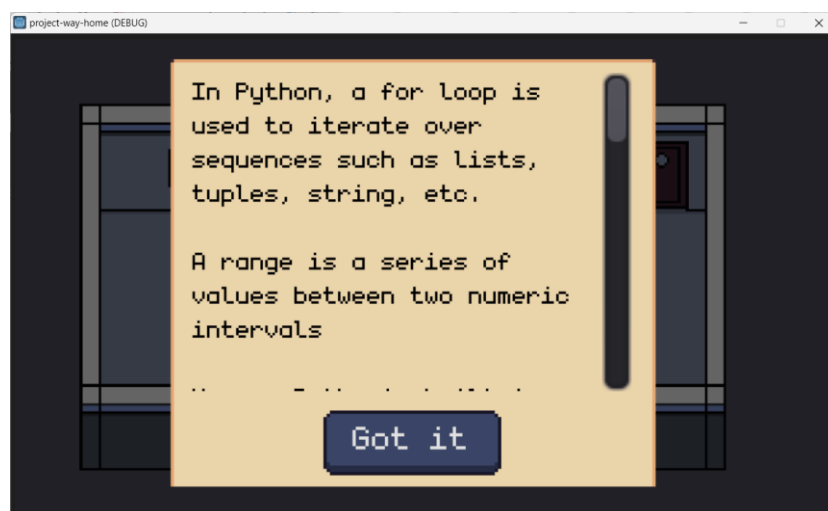


Figure 22: Level 5 – Loops Topic

Figure 22 shows the Level 5 – Loop Layout. The user gets prompts about the topic on level 5 about input.

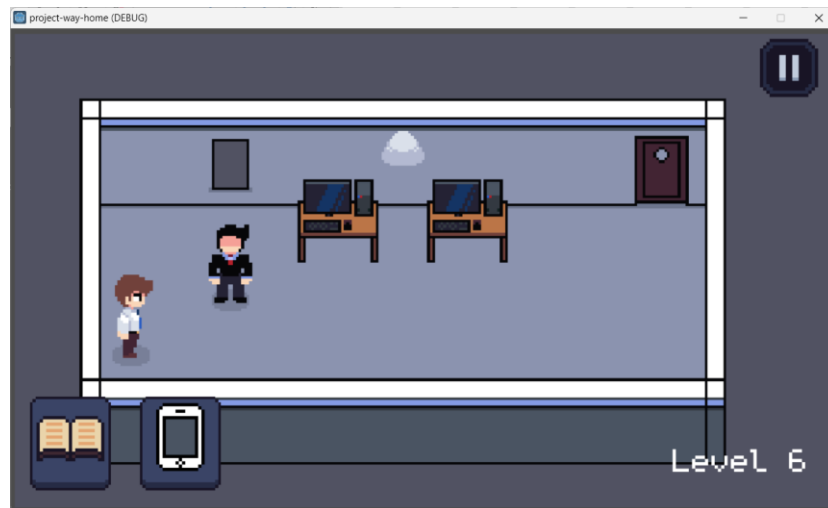


Figure 24: Level 6 – Lists Layout

Figure 24 shows the Level 6 – Lists Layout. The user sees this layout of the level upon entering level 6 of the game.

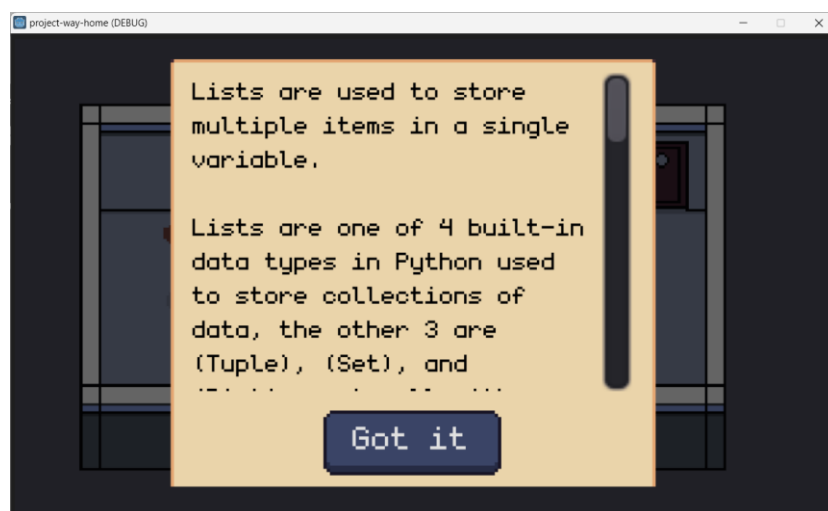


Figure 25: Level 6 – Lists Topic

Figure 25 shows the Level 6 – Lists Topic. The user gets prompts about the topic on level 6 about Lists

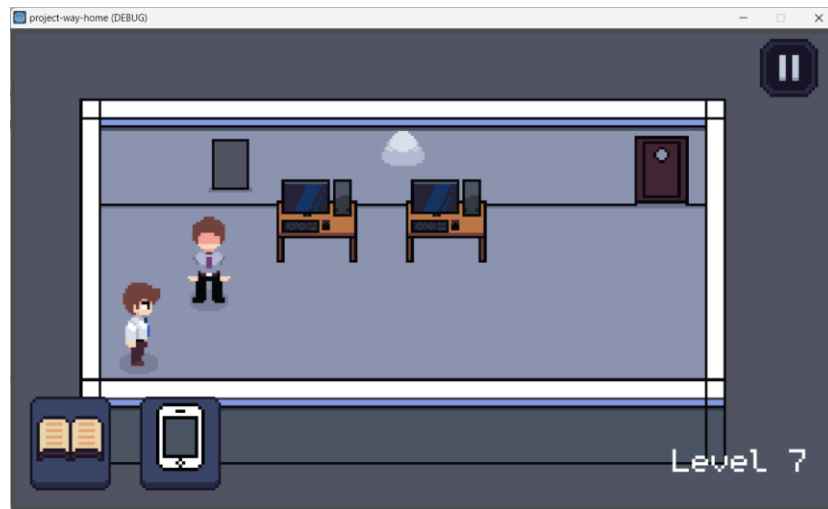


Figure 26: Level 7 – Dictionary Layout

Figure 26 shows the Level 7 – Dictionary Layout. The user sees this layout of the level upon entering level 7 of the game.

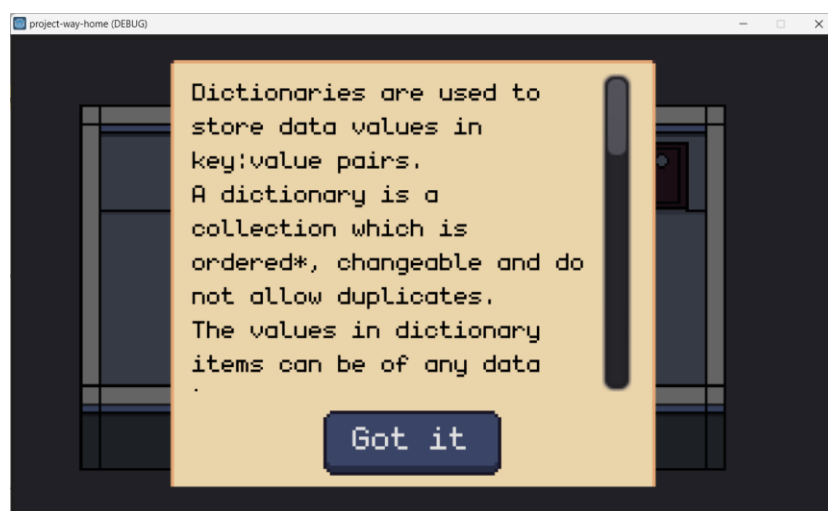


Figure 27: Level 7 – Dictionary Topic

Figure 27 shows the Level 7 – Dictionary Topic. The user gets prompts about the topic on level 7 about Lists



Figure 28: Level 8 – Function Layout

Figure 28 shows the Level 8 – Function Layout. The user sees this layout of the level upon entering level 8 of the game.

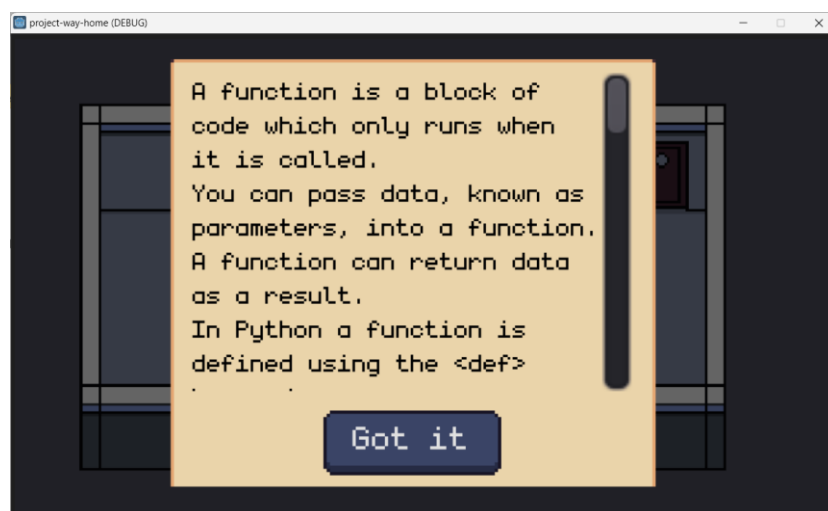


Figure 29: Level 8 – Function Topic

Figure 29 shows the Level 8 - Function Topic. The user gets prompts about the topic on level 8 about Functions



Figure 30: Game Assessment Screen

Figure 30 shows the Game's Final Assessment. The user can access the Game's Final Assessment by completing the final level, 8, wherein the user gets a randomized set of 10 questions about the topics learned from the game. The user can see the texts that shows the number of questions, the total score, the question asked and the user can select three buttons that acts as their answers when selected by the user.



Figure 31: Game Assessment Score Screen

Figure 31 shows the Game's Final Assessment Score screen that is shown to the user after answering the set of 10 questions given by the game's final assessment. A button labeled menu is on the center that redirects the user back to the main menu when pressed.

Discussion of Methodology Phases

Python programming simulation through videogames aims to provide the students to learn python programming by videogames as a mean of learning. The criteria for the evaluation included the characteristics defined as functionality, reliability, usability and portability.

Planning and Requirements. In this phase, the researchers determined the development sequence. They identified the list of features that would be implemented on the videogame and determine the scope whilst producing a development plan all around it.

Analysis and Design. In designing the feature, the researchers carried out development on creating design for the videogame such as the game modules and the videogame interface as a whole.

Implementation. In this phase, the researchers modified the videogame to implement features to organize and analyze the functionality of the videogame with the users.

Testing and Evaluation. In this phase, the researchers produced functionality in features in accordance to the planned videogame and the usage of the iterative development. This was the time where the researchers conducted code inspections.

System Software Evaluation Results

The researchers conducted an evaluation to the selected participants composed of IT Students, CS Students and Others.

Table 3: General rating of the respondent's evaluation of the game in terms of functionality

DESCRIPTION	MEAN	INTERPRETATION
1. The simulation runs responsively and accurately	4.8	EXCELLENT
2. The learning module provides accurate information	4.96	EXCELLENT
3. The player movement is responsive and accurate	5	EXCELLENT
4. The compiler determines the error on the simulation	3.6	VERY GOOD
Weighted Mean	4.59	EXCELLENT

The table above shows the general rating of the respondents' evaluation of the game in terms of functionality. The mean score obtained from description 1 was 4.8 which is interpreted as Excellent. This implies that the system is responsive. For description 2, the mean score was 4.96 which is also interpreted as Excellent and this implies that the learning module provides accurate information. For description 3, the mean score obtained was 5 and interpreted as Excellent which shows that the player movement is responsive and accurate. For description 4, the mean score obtained was 3.66 which is interpreted as Very Good and it implies that the system can somewhat determines the error on the simulation. In general, the quality characteristics of functionality gained a total mean of 4.59 which is interpreted as Excellent and this implies that the system's functionality gained above average for the satisfaction of the users.

Table 4: General rating of the respondents' evaluation of the game in terms of usability.

DESCRIPTION	MEAN	INTERPRETATION
1. The game's diary and simulation display clarity	4.8	EXCELLENT
2. The game runs smoothly	4.56	EXCELLENT
3. The user interface is user friendly	4.96	EXCELLENT
Weighted Mean	4.77	EXCELLENT

The table above represents the general rating of the respondents' evaluation of the game in terms of usability. For description 1, the mean score was 4.8 and interpreted as Excellent which only shows that the game's diary and simulation displays clarity. For description 2, the mean score was 4.56 which is interpreted as Excellent and this implies that the game is smooth running. Lastly for Description 3, the mean score was 4.96 and interpreted as Excellent which shows that the system interface is user-friendly. In general, the quality characteristic of usability gained a total mean of 4.77 which is also interpreted as Excellent and implies that the system's usability gained above average for the satisfaction of the users.

Table 5. General rating of the respondents' evaluation of the system in terms of reliability.

DESCRIPTION	MEAN	INTERPRETATION
1. The game provides accurate assessment	3.6	VERY GOOD
2. The game displays the level progression accurately	4..84	EXCELLENT
3. The tutorial is clear and concise	4	VERY GOOD
Weighted Mean	4.14	VERY GOOD

The table above represents the general rating of the respondents' evaluation of the system in terms of reliability. For description 1, the mean score was 3.6 and interpreted as Very Good which only implies that the system provides accurate assessment. For Description 2, the mean score was 4.84 and interpreted as Excellent which only that the game displays the level progression accurately. For Description 3, the mean score was 4 and interpreted as Very Good which only shows that the tutorial is clear and concise. In general, the quality characteristics of reliability gained a total mean score of 4.14 which is interpreted as Very Good. This implies that the system's reliability gained average for the satisfaction of the users.

Table 6: General rating of the respondents' evaluation of the system in terms of efficiency.

DESCRIPTION	MEAN	INTERPRETATION
1. The game responds promptly to user commands	3.64	VERY GOOD
2. The system is complete and runs without any errors	3.2	GOOD
3. The instruction of the game are straightforward	3.16	GOOD
Weighted Mean	3.33	GOOD

The table above represents the general rating of the respondents' evaluation of the system in terms of efficiency. For description 1, the mean score was 3.64 and interpreted as Very Good which shows that the game responds promptly to user commands. For description 2, the mean score obtained was 3.2 and interpreted as Good which implies that the system is complete and somewhat runs without any errors. For description 3, the mean score was 3.16 and interpreted as Good which shows that the instruction of the game is somewhat straightforward. In general, the quality characteristics of efficiency gained a total mean score

of 3.33 and interpreted as Good. This implies that the system's efficiency gained somewhat average for the satisfaction of the users.

Table 7: General rating of the respondents' evaluation of the system in terms of portability.

DESCRIPTION	MEAN	INTERPRETATION
1. The game is easy to install	4.8	EXCELLENT
2. The game can adapt to software requirements	4.8	EXCELLENT
3. The game is easy to navigate	4.8	EXCELLENT
Weighted Mean	4.8	EXCELLENT

The table above represents the general rating of the respondents' evaluation of the system in terms of portability. For description 1, the mean score obtained was 4.8 which is interpreted as Excellent and this implies that the game is easy to install. For description 2, the mean score obtained was 4.8 which is interpreted as Excellent and this implies that the game can adapt to software requirements. For description 3, the mean score obtained was also 4.8 which is interpreted as Excellent and this implies that the game is easy to navigate. In general, the quality characteristics of portability gained a total mean score of 4.8 which is interpreted as Excellent. This implies that the system's portability gained above average for the satisfaction of the users.

Table 8: Overall Rating of the System.

DESCRIPTION	MEAN	INTERPRETATION
FUNCTIONALITY	4.59	EXCELLENT
USABILITY	4.77	EXCELLENT
RELIABILITY	4.14	VERY GOOD
EFFICIENCY	3.33	GOOD
PORTABILITY	4.8	EXCELLENT
Weighted Mean	4.32	VERY GOOD

The table above represents the overall rating of the respondents' evaluation of the characteristics of the system. The overall mean score for the functionality is 4.59 and interpreted as Excellent which only implies that the system's functionality gained above average for the satisfaction of the users. The usability's overall mean score is 4.77 and interpreted also as Excellent which only implies that the system's usability has gained above average for the satisfaction of the users. The overall mean score for the reliability is 4.14 and interpreted as Very Good which implies that the system's reliability has gained average for the satisfaction of the users. The efficiency's overall mean score is 3.33 and interpreted as Good which only implies that the system's efficiency has gained somewhat average for the satisfaction of the users. Lastly, the overall mean score for the portability is 4.8 and interpreted also as Excellent which only implies that the system's portability has gained above average for the satisfaction of the users.

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

This chapter provides an overall summary of the research conducted by the researchers. Furthermore, it includes the conclusion and recommendation. the future development and enhancement of this study.

Summary

The study entitled Game Simulation for Alternative Learning in Basic Python is for students who are taking the courses under Computer Science and Information Technology

The study is focused on creating a game application wherein students can interact and learn Python Programming Language using the game simulation. Also, this application can be used by faculty as an alternative method in teaching Python Programming Language. The researchers employed a variety of data collection methods to acquire data to support the envisioned application. The survey was conducted with the target client, in order to determine the issues that needed to be rectified. Significant information was gathered, primarily through internet journals and other similar research published online.

The development of the proposed application was undergone according to the following process of evolutionary model: phase 1: planning and requirements; phase 2: analysis and design; phase 3: implementation; phase 4: testing; and phase 5: evaluation and review. This application was built using the Godot Game Engine in where it is used in encoding and editing, although the researchers utilized Github for code hosting tools for repositories management and collaboration. Aseprite and Photoshop were used to modify and create art assets that is used in the game.

In this study, evaluation was performed. The researchers utilized a convenient sampling method. The convenience sampling is a non-probability sampling technique that involves selecting participants based on their easy availability and accessibility. Due to the researchers chosen participants, there are 3 “other” respondents, 26 CS Students, and 23 IT

Students. The total mean is 4.32, with the equivalent of Very Good, indicating that the system is average for what it offers. In getting the rating scale of the application, the researchers used the Likert scale method.

Conclusion

As proven, the developed application made for the study entitled Game Simulation for Alternative Learning in Basic Python is capable of meeting the demands needed for the users; students and faculties, by offering the capability of learning the Python Programming Language in more enjoyable way, by providing alternatives and giving enough lesson modules and assessments so that at the end of the game, the researchers were able to know if the user understands the topics. Researchers identified the subjects in order to determine whether or not the system is effective as an alternative learning method. For the study, the researchers employed convenience sampling, and there are 3 “other” respondents, 26 CS Students, and 23 IT students. The overall mean is 4.32, with the equivalent of Very Good, meaning that the system is average and viable as an alternative learning method on learning basic python programming language

Recommendations

Although the application is able to satisfy the needs of the client and potential users, subjects stated that there are several areas that may be enhanced. According to the participant's, organization's, and professionals' suggestions, the researchers recommend that:

1. To strengthen the research design and enhance the validity and generalizability of the findings, researchers may consider complementing convenience sampling with other sampling methods, such as random sampling or stratified sampling, to ensure a more representative and diverse participant pool.

2. Implement much more levels and more understandable lessons and modules that would teach basic python much more efficiently.

3. Provide support on multiple platforms such as mobile and other devices to make the system accessible and convenient.

REFERENCES

- Qian, M., & Clark, K. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63, 50–58.
<https://doi.org/10.1016/j.chb.2016.05.023>
- Hitchens, M., & Drachen, A. (2008). The Many Faces of Role-Playing Games. *International Journal of Role-Playing*, (1), 3–21. <https://doi.org/10.33063/ijrp.vi1.185>
- Dimitrios Vlachopoulos, & Agoritsa Makri. (2017). The effect of games and simulations on higher education: a systematic literature review. 14(1). <https://doi.org/10.1186/s41239-017-0062-1>
- Stack Overflow Developer Survey 2022. (n.d.). Stack Overflow.
<https://survey.stackoverflow.co/2022/#most-popular-technologies-language>
- Coursera. (2023). What Is Python Used For? A Beginner's Guide. Coursera.
<https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- Fran. (2021, December 30). What is Python used for? | 10 practical Python uses - FutureLearn. FutureLearn.
<https://www.futurelearn.com/info/blog/what-is-python-used-for>
- Python, R. (2023). 11 Beginner Tips for Learning Python Programming. [realpython.com](https://realpython.com/python-beginner-tips/).
<https://realpython.com/python-beginner-tips/>
- Taylor, S. (2023). Python Data Structures. Corporate Finance Institute.
<https://corporatefinanceinstitute.com/resources/data-science/python-data-structures/#:~:text=The%20basic%20Python%20data%20structures,group%20data%20according%20to%20type.>

Moving from Waterfall to Iterative Development: An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP. (2011, August 1). IEEE Conference Publication | IEEE Xplore.

<https://ieeexplore.ieee.org/abstract/document/6068383/references#references>

Gestwicki, P. (2021). Godot engine and checklist-based specifications: revising a game programming class for asynchronous online teaching. *Journal of Computing Sciences in Colleges*, 37(4), 30-40.

<http://www.ccsc.org/publications/journals/MW2021.pdf#page=30>

Pritchard, D. E., & Vasiga, T. (2013). CS circles.

<https://doi.org/10.1145/2445196.2445370>

Huang, X., Zhang, H., & Yin, S. (2021, May). Using Digital Game-Based Language Learning for Foreign Language Learning: a Scoping Review on Previous Studies Published in between 2015 and 2020. In *DPU International Conference on Business Innovation and Social Sciences 2021* (pp. 434-442). Dhurakij Pundit University.

<https://www.airitilibrary.com/Publication/alDetailedMesh?docid=P20210713001-202105-202107130016-202107130016-434-442>

Liaskovych, D., & Tarasova, O. (2021). GAME AS A METHOD OF THE EDUCATIONAL PROCESS INTENSIFICATION IN FOREIGN LANGUAGE TEACHING. *МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ*, 39.

<https://researcheurope.org/wp-content/uploads/2021/01/re-22.01.2021.pdf#page=39>

- Upadhyay, S. (2023). What is Backtracking Algorithm? Types, Examples & its Application. Simplilearn.com.<https://www.simplilearn.com/tutorials/data-structure-tutorial/backtracking-algorithm#:~:text=Backtracking%20is%20a%20general%20algorithm,completed%20to%20a%20reasonable%20solution>.
- Bodie, J. (2023, February 3). Graphic Design In The Video Gaming Industry (Complete Guide) | Medium. Medium. <https://medium.com/@jasonrbodie/graphic-design-in-the-video-gaming-industry-complete-guide-daeb0d07ea84>
- Holmgren, H. (2016). Animated vs. Static graphics in a video game | Linköping University. <https://www.diva-portal.org/smash/get/diva2:1040895/FULLTEXT01.pdf>
- Zufri, T., Hilman, D., & Frans, O. (2022). Research on the Application of Pixel Art in Game Character Design. *Journal of Games, Game Art and Gamification*, 7(1), 27–31. <https://doi.org/10.21512/jggag.v7i1.8565>

APPENDICES

Appendix 1. Time Table of Activities

MAJOR ACTIVITIES	IMPLEMENTATION				
	MARCH	APRIL	MAY	JUNE	JULY
1. Inception					
Interview					
Consultation					
2. Elaboration					
Construct Diagrams					
Refining Modules					
3. Construction					
Program Coding					
Research Documentation					
4. Transition					
Code Inspection					
System Evaluation					

Appendix 2. System Evaluation Form



Republic of the Philippines
CAVITE STATE UNIVERSITY
 Imus Campus
 Cavite Civic Center Palico IV, Imus, Cavite
 Telefax: (046) 471-66-07 / (046) 471-67-70 / (046) 686-23-49
www.cvsu.edu.ph

GAME SIMULATION FOR ALTERNATIVE LEARNING IN BASIC PYTHON

NAME: (OPTIONAL): _____ DATE: _____

☐ CS Student

☐ IT Student

☐ Other

Instructions: Kindly evaluate the application by using the given scale and placing a checkmark (✓) under the corresponding numerical rating:

RATING	EQUIVALENT	DESCRIPTION
5	Excellent	The system is acceptable and exceeds expectation
4	Very Good	The system meets all the expectation
3	Good	The system meets most of its expectation
2	Fair	The system barely meets its expectation
1	Poor	The system did not meet the expectation

FUNCTIONALITY	1	2	3	4	5
The simulation runs responsively and accurately					
The learning module provides accurate information.					
The player movement is responsive and accurate					
The compiler determines the error on the simulation.					

USABILITY	1	2	3	4	5
The game's diary and simulation display clarity.					
The game runs smoothly.					
The user interface is user-friendly					

RELIABILITY	1	2	3	4	5
The game provides accurate assessment					
The game displays the level progression accurately.					
The tutorial is clear and concise.					

EFFICIENCY	1	2	3	4	5
The game responds promptly to user commands.					
The system is complete and runs without any errors.					
The instructions of the game are straightforward.					

PORTABILITY	1	2	3	4	5
The game is easy to install.					
The game can adapt to software requirements.					
The game is easy to navigate					

Comments/Suggestions:

(Evaluator's signature)

Validated by:



GRACE S. IBANEZ
Thesis Adviser



KLAIID BENDIO MORAN
Technical Adviser

Appendix 3. Curriculum Vitae

Khaizter Vash Topacio

Software Developer

Phone: (+63) 9772181373

Email: khaizter.vashh@gmail.com

Location: Cavite, Philippines

Portfolio: <https://khaizter-developer-portfolio.netlify.app/>

GitHub: <https://github.com/khaizter>

LinkedIn: <https://www.linkedin.com/in/khaizter-vash-topacio-a22bb0232/>

Professional Profile

A passionate Software Developer having a special interest in Frontend technologies and experience of building Web applications with JavaScript / ReactJS / NodeJS and some other cool libraries and frameworks.

Skills

Front-end

- React
- React router
- Redux
- Styled Components
- SASS
- Material UI
- Tailwind CSS
- Framer Motion

Back-end

- Node JS
- Express
- REST API
- Mongo DB
- MySQL

Others

- Git
- GitHub
- Figma
- Typescript

Projects

>To-do App

A full stack application built using MERN Stack, during development I learn a lot of things on the backend side like authentications, JSON web tokens, CORS, and etc. I also used other libraries like formic and yup in form validations combined with express validation in backend.

Features

- Add, delete, update status, reorder task
- Filtering of task by status
- Authentication
- Theme switching (light mode, dark mode)

Live Demo: <https://todo-khaizter.netlify.app/>

Repository: <https://github.com/khaizter/todo-app>

>My Movie List

A front-end application that uses TMDB API, during development I learned a lot of things like fetching data in a REST API, and also code reusable components like pagination, modals, dropdowns and also work with react router.

Features

- Viewing specific movie details
- Viewing trailers
- Movie filtering by genre or search query

Live Demo: <https://khaizter-my-movie-list.netlify.app/>

Repository: <https://github.com/khaizter/my-movie-list>

>Front-end application using Country API

In this website I applied the things I learn from using styled components like theme switching and also playing with animations and transitions using framer motion.

Features

- Viewing country details
- Filtering country by region and search query
- Theme switching (light mode, dark mode)

Live Demo: <https://khaizter-where-in-the-world.netlify.app/>

Repository: <https://github.com/khaizter/country-theme-switch>

>Sneakers Ecommerce

In this website I applied the things I learned in building responsive page and using React hooks, during development I also learned to build reusable UI/UX components. I also applied the things I learned from organizing the project structures like folders and use BEM methodology.

Features

- Adding and removing items from cart
- Viewing products via light box

Live Demo: <https://khaizter.github.io/ecommerce-product-page/>

Repository: <https://github.com/khaizter/ecommerce-product-page>

Educational Profile

University of Perpetual Help (2017 – 2019)

Information, Communication and Technology (ICT)

Cavite State University (2019 – Present)

Bachelor Science in Computer Science (BSCS)



RONNETH JOY GERVACIO

Contact

84-E Aniban 4 Bacoor Cavite, 4102

0977-1916-387

ronnethjoygrvc@gmail.com

Skills

Basic knowledge in CSS, HTML, PHP

Surface knowledge in Microsoft Office

Proficiency in essay writing

Social Media Knowledge

Verbal & Written Communication

Leadership & Teamwork

Affiliations

Supreme Student Government (SSG)
2013 - 2017

Student Teachers in Senior High School
2019

Honor Society School Organization
2022 - 2023

Language

English ●●●●●●●●

Filipino ●●●●●●●●

About Me

An independent woman who is dedicated, detail-oriented, and self-motivated. I consider myself a responsible and orderly person. I am able to adapt easily and have a good communication skills.

Education

Bacoor National High School
Junior High School
2013 - 2017

Asian Institute of Computer Studies
Senior High School
2017 - 2019

Cavite State University - Imus Campus
Bachelor of Science in Computer Science
2019 - Present

Work Experience

Commission on Elections, Office of the Election Officer, Bacoor City, Cavite
November 2018 - January 2019
Intern

- Process the voter's applications
- Receive and respond to questions from the public with regard to voter registration and the voting process.

McDonald's Golden Food Archers Inc.
Bacoor City, Cavite
March 2019 - July 2019
Customer Service Crew

- Takes orders and answers questions on meal items. Makes recommendations and serves food/beverages to customers. Prepares the bill that itemizes total meal costs and sales taxes.

Melham Construction Corporation
Quezon City
July 2022 - August 2022
Intern

- Assigned in maintenance team that handles the company's website. Handles debugging and coding of the system

VXI Global Solutions, LLC
Pasay City
September 2022 - February 2023
Customer Service Representative (CSR)

- Identify and assess customers' needs to achieve satisfaction.
- Provide accurate, valid, and complete account, product, and service information by using the right methods or tools.
- Resolve product or service problems by clarifying customers' complaints.



Glenn Ace Liwmell Gargar

Computer Science Student

Profile

Address

Block 1 Lot 8 Greenside Homes
Malagasang II-D
Imus, Cavite

Phone

09477427796

Email Address

itsmeglenngar@gmail.com

Skills

- Basic Knowledge with Java, Python, PHP, HTML, MySQL
- Basic Knowledge with Adobe Photoshop CS 6
- Proficiency in Microsoft Office applications (Word, PowerPoint, Excel)
- Reps are on the front lines of satisfaction and takes responsibility

Languages

English C2



Filipino



To conquer my confidence and my fears in the field of job and obtain a position in a reputable organization wherein I can fully utilize my educational and professional skills in the field of information technology and experience new things while working well with different people and contributing to the success of the company.

Work Experience

Quick Service Restaurants (QSR) Corporation - Imus (March 2019 - June 2019)

Service Crew

Part-time summer job. I learned about more customer service and improved my communication skills with customers and co-workers. Understood more about following safety procedures and to safely use equipments and supplies.

Quick Service Restaurants (QSR) Corporation - Imus (March 2018 - June 2018)

Service Crew

Part-time summer job. I gained some knowledge of food preparation and presentation methods, techniques, and quality standards. Was able to follow routine verbal and written instructions.

Education

Cavite State University – Imus Campus **Tertiary**
Bachelor of Science in Computer Science (August 2019 - Present)

College

Kin Yang Academy **Secondary**
TVL - Home Economics (June 2017 - March 2019)

Senior High School

Kin Yang Academy **Secondary**
(June 2013 - March 2017)

Malagasang II Elementary School **Primary**
(June 2008 - March 2013)

Rafael Palma Elementary School **Primary**
(June 2007 - March 2008)

Personal Information

Date of Birth
January 18, 2002

Place of Birth
Ospital ng Maynila

Gender
Male

Age
20

Height / Weight
5'6" / 72kgs.

Status
Single

Appendix 4. Sample Code

PLAYER MOVEMENT.gd

```

extends KinematicBody2D

signal player_interact()

onready var animation_player = $animation_player
onready var sprite = $sprite

export var move_speed = 10

var velocity = Vector2()
var moving = velocity.normalized().length() != 0
var facing_direction = velocity.normalized()

func _ready():
    pass

func _input(event):
    if (event.is_action_pressed("interact")):
        interact()

func _physics_process(delta):
    var input_vector = Vector2(float(Input.is_action_pressed("move_right")) -
float(Input.is_action_pressed("move_left")),
float(Input.is_action_pressed("move_down")) -
float(Input.is_action_pressed("move_up")))
    velocity = input_vector * move_speed

    update_animation(input_vector)
    velocity = move_and_slide(velocity)
#fluidity
func update_animation(input_vector):

    var moving = input_vector.length() != 0 and is_physics_processing()

    if (input_vector.normalized() == Vector2.DOWN):
        facing_direction = Vector2.DOWN
    elif (input_vector.normalized() == Vector2.UP):
        facing_direction = Vector2.UP
    elif (input_vector.normalized() == Vector2.RIGHT):
        facing_direction = Vector2.RIGHT
    elif (input_vector.normalized() == Vector2.LEFT):
        facing_direction = Vector2.LEFT

    if (facing_direction == Vector2.DOWN):

```

```

        sprite.scale.x = 1
        animation_player.play("walk_down" if moving else "idle_down")
    elif (facing_direction == Vector2.UP):
        sprite.scale.x = 1
        animation_player.play("walk_up" if moving else "idle_up")
    elif (facing_direction == Vector2.RIGHT):
        sprite.scale.x = 1
        animation_player.play("walk_side" if moving else "idle_side")
    elif (facing_direction == Vector2.LEFT):
        sprite.scale.x = -1
        animation_player.play("walk_side" if moving else "idle_side")
#controls
func freeze():
    set_physics_process(false)
    update_animation(facing_direction)

func unfreeze():
    set_physics_process(true)
    update_animation(facing_direction)

func interact():
    emit_signal("player_interact")

```

MAIN MENU.gd

```

extends CanvasLayer

export var level_1_scene:PackedScene
export var level_2_scene:PackedScene
export var level_3_scene:PackedScene
export var level_4_scene:PackedScene
export var level_5_scene:PackedScene
export var level_6_scene:PackedScene
export var level_7_scene:PackedScene
export var level_8_scene:PackedScene

onready var main = $control/main
onready var stages = $control/stages
onready var level_1 = $control/stages/level_1"
onready var level_2 = $control/stages/level_2"
onready var level_3 = $control/stages/level_3"
onready var level_4 = $control/stages/level_4"
onready var level_5 = $control/stages/level_5"
onready var level_6 = $control/stages/level_6"
onready var level_7 = $control/stages/level_7"
onready var level_8 = $control/stages/level_8"

```

```

func _ready():
    init()

func init():
    level_1.disabled = not Player.is_level_unlocked("1")
    level_2.disabled = not Player.is_level_unlocked("2")
    level_3.disabled = not Player.is_level_unlocked("3")
    level_4.disabled = not Player.is_level_unlocked("4")
    level_5.disabled = not Player.is_level_unlocked("5")
    level_6.disabled = not Player.is_level_unlocked("6")
    level_7.disabled = not Player.is_level_unlocked("7")
    level_8.disabled = not Player.is_level_unlocked("8")
    main.visible = true
    stages.visible = false

func _on_start_button_pressed():
    SoundMaster.play("menu_confirm")
    main.visible = false
    stages.visible = true

func _on_exit_button_pressed():
    get_tree().quit()

func _on_back_pressed():
    SoundMaster.play("menu_back")
    main.visible = true
    stages.visible = false

func _on_level_1_pressed():
    SoundMaster.play("menu_confirm")
    if (level_1_scene):
        get_tree().change_scene_to(level_1_scene)

func _on_level_2_pressed():
    SoundMaster.play("menu_confirm")
    if (level_2_scene):
        get_tree().change_scene_to(level_2_scene)

func _on_level_3_pressed():
    SoundMaster.play("menu_confirm")
    if (level_3_scene):
        get_tree().change_scene_to(level_3_scene)

func _on_level_4_pressed():
    SoundMaster.play("menu_confirm")
    if (level_4_scene):
        get_tree().change_scene_to(level_4_scene)

```

```

func _on_level_5_pressed():
    SoundMaster.play("menu_confirm")
    if (level_5_scene):
        get_tree().change_scene_to(level_5_scene)

func _on_level_6_pressed():
    SoundMaster.play("menu_confirm")
    if (level_6_scene):
        get_tree().change_scene_to(level_6_scene)

func _on_level_7_pressed():
    SoundMaster.play("menu_confirm")
    if (level_7_scene):
        get_tree().change_scene_to(level_7_scene)

func _on_level_8_pressed():
    SoundMaster.play("menu_confirm")
    if (level_8_scene):
        get_tree().change_scene_to(level_8_scene)

func _on_reset_button_pressed():
    SoundMaster.play("menu_back")
    Player.reset_data()
    init()

func _on_credits_button_pressed():
    SoundMaster.play("menu_back")
    get_tree().change_scene("res://levels/end_credits/end_credits.tscn")

```

GAME INTERFACE.gd

```

extends CanvasLayer

signal freeze()
signal unfreeze()

export(Array,String, MULTILINE) var notes_data

export(String, MULTILINE) var tutorial_text
export(String, MULTILINE) var initial_code
export(String, MULTILINE) var checker

onready var main_layer = $main

```



```

onready var menu_layer = $menu
onready var diary_layer = $diary
onready var phone_layer = $phone

onready var page_layer = $diary/page_selector
onready var notes_layer = $diary/notes_layer
onready var notes = $diary/notes_layer/notes

onready var page_1_button = $diary/page_selector/page_1
onready var page_2_button = $diary/page_selector/page_2
onready var page_3_button = $diary/page_selector/page_3
onready var page_4_button = $diary/page_selector/page_4
onready var page_5_button = $diary/page_selector/page_5
onready var page_6_button = $diary/page_selector/page_6
onready var page_7_button = $diary/page_selector/page_7
onready var page_8_button = $diary/page_selector/page_8

onready var code_editor = $phone/code_editor
onready var code_feedback = $phone/code_feedback

onready var level = $main/level

const SCRIPT_FILE = "res://script.py"

const SCRIPT_TEMPLATE = "import sys\n\ndef
eval():\n\ttry:\n\t\t#initial\n\t\t#code\n\t\t#checker\n\t\texcept Exception as
error:\n\t\t\treturn [type(error).__name__ + '-' + str(error),False]\n\tres =
eval()\n\tsys.stdout.write(res[0] + '^' + str(res[1]))"

func _ready():
    main_layer.visible = true
    menu_layer.visible = false
    diary_layer.visible = false
    phone_layer.visible = false

    page_layer.visible = true
    notes_layer.visible = false

    page_1_button.disabled = not Player.is_level_unlocked("1")
    page_2_button.disabled = not Player.is_level_unlocked("2")
    page_3_button.disabled = not Player.is_level_unlocked("3")
    page_4_button.disabled = not Player.is_level_unlocked("4")
    page_5_button.disabled = not Player.is_level_unlocked("5")
    page_6_button.disabled = not Player.is_level_unlocked("6")
    page_7_button.disabled = not Player.is_level_unlocked("7")
    page_8_button.disabled = not Player.is_level_unlocked("8")

```

```
func set_level_text(value):
    level.text = 'Level ' + str(value)

func _on_menu_pressed():
    SoundMaster.play("menu_confirm")
    main_layer.visible = false
    menu_layer.visible = true
    emit_signal("freeze")

func _on_main_menu_button_pressed():
    SoundMaster.play("menu_confirm")
    get_tree().change_scene("res://levels/main_menu/main_menu.tscn")

func _on_exit_game_button_pressed():
    SoundMaster.play("menu_back")
    get_tree().quit()

func _on_continue_button_pressed():
    SoundMaster.play("menu_back")
    main_layer.visible = true
    menu_layer.visible = false
    emit_signal("unfreeze")

func _on_diary_button_pressed():
    SoundMaster.play("menu_confirm")
    main_layer.visible = false
    diary_layer.visible = true
    emit_signal("freeze")

func _on_close_diary_button_pressed():
    SoundMaster.play("menu_back")
    main_layer.visible = true
    diary_layer.visible = false
    emit_signal("unfreeze")

func _on_phone_button_pressed():
    SoundMaster.play("menu_confirm")
    main_layer.visible = false
    phone_layer.visible = true
    emit_signal("freeze")

func _on_close_phone_button_pressed():
    SoundMaster.play("menu_back")
    main_layer.visible = true
    phone_layer.visible = false
    emit_signal("unfreeze")
```

```
func _on_page_1_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[0]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_2_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[1]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_3_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[2]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_4_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[3]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_5_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[4]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_6_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[5]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_7_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[6]
    page_layer.visible = false
    notes_layer.visible = true

func _on_page_8_pressed():
    SoundMaster.play("page_confirm")
    notes.text = notes_data[7]
    page_layer.visible = false
    notes_layer.visible = true
```

```

func _on_back_to_page_button_pressed():
    SoundMaster.play("page_back")
    page_layer.visible = true
    notes_layer.visible = false

func _on_run_phone_button_pressed():
    var result = evaluate_code(code_editor.text).split("^")
    print("test ",result)
# emit_signal("finish_problem",result[0], result[1] == 'True')
    if "default_output" in result[0]:
        code_feedback.text = result[0].replace('default_output','')
        pass
    else:
        code_feedback.text = result[0]

func _on_reset_phone_button_pressed():
    code_editor.text = ''

func evaluate_code(input):
    # do transformations on user script and check script

    var initial_script = initial_code
    initial_script = '\t\t' + initial_script.replace("\n","\n\t\t")

    var regex = RegEx.new()
    regex.compile("input\\(.*\\)")

    var user_script = input
    user_script = "exec(''" + regex.sub(user_script.replace("\'", "\""),
    "input-field",true) + "'",globals(),ldict)"

    var check_script = checker
    check_script = '\t\t' + check_script.replace("\n","\n\t\t")

    # edit script.py
    var file = File.new()
    file.open(SCRIPT_FILE,File.WRITE)
    var new_script = SCRIPT_TEMPLATE.replace("#code",
user_script).replace("#checker",check_script).replace("#initial",initial_scrip
t)
    file.store_string(new_script)
    file.close()

    # evaluate code
    var global_dir_path = ProjectSettings.globalize_path("res://")
    var stdout = []

```

```

    var exit = OS.execute("python",[global_dir_path + "/script.py"],true ,
stdout,true)
    return stdout[0]

```

LEVEL1.gd

```

extends Node2D

onready var door = $door

onready var tutorial_layer = $tutorial_layer
onready var player = $ysort/player
onready var game_interface = $game_interface
onready var menu_confirm_sfx = $menu_confirm_sfx

onready var npc = $ysort/new_npc_1
onready var computer = $ysort/new_computer_1

var task_1 = false

func _init():
    pass

func _ready():
    player.freeze()
    tutorial_layer.visible = true
    game_interface.visible = false
    game_interface.set_level_text(1)

func _process(delta):
    if task_1:
        door.open = true
        Player.unlock_level("2")

# game interface
func _on_game_interface_freeze():
    player.freeze()

func _on_game_interface_unfreeze():
    player.unfreeze()

```

```
# tutorial_layer
func _on_continue_pressed():
    menu_confirm_sfx.play()
    tutorial_layer.visible = false
    game_interface.visible = true
    player.unfreeze()

# npc
func _on_new_npc_1_start_solving():
    computer.enable()

func _on_new_npc_1_freeze_player():
    player.freeze()
    game_interface.visible = false

func _on_new_npc_1_unfreeze_player():
    player.unfreeze()
    game_interface.visible = true

# computer
func _on_new_computer_1_quest_done():
    task_1 = true
    npc.set_dialogue_page(1)

func _on_new_computer_1_freeze_player():
    player.freeze()
    game_interface.visible = false

func _on_new_computer_1_unfreeze_player():
    player.unfreeze()
    game_interface.visible = true
```