OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Pao Tycoon: A Language Study for Lua Scripting

Kuo-pao Yang[1], Aaron Madison[2], Bailey Milburn[3], Steele Russell[4], Zakkary Huckab[5]

[1, 2, 3, 4, 5]*Computer Science Department, Southeastern Louisiana University, Hammond, Louisiana, USA*

ABSTRACT: *This paper presents an exploration of the Lua programming language through the design and development of a Roblox-based tycoon game titled Pao Tycoon. Lua, a lightweight and embeddable scripting language, has become widely adopted in game development, software customization, and numerous modern applications. In this study, we examine Lua's historical origins, key design influences, and defining language features, emphasizing its simplicity, flexibility, and expressive scripting capabilities. Using Pao Tycoon as a case study, we demonstrate how Lua's dynamic typing, concise syntax, and extensible architecture support rapid prototyping, modular game mechanics, and effective collaborative development within Roblox Studio. This work highlights Lua's continued relevance and illustrates how its lightweight yet powerful design makes it well suited for both educational environments and professional game development.*

KEY WARDS: *Lua Programming; Roblox; Game Design*

Date of Submission: 15-11-2025 Date of acceptance: 30-11-2025

I. INTRODUCTION

This study focuses on the design and development of a tycoon-style game within Roblox, using Lua to implement complex game mechanics and interactive elements. In particular, we explore the use of Lua within Roblox Studio to build a game called Pao Tycoon, which highlights the language's ability to handle real-time multiplayer interactions, dynamic environments, and custom game features. By examining how Lua is leveraged for both the creation of game mechanics and the management of backend functions, this report provides insight into the benefits and applications of Lua in modern game development. Additionally, it emphasizes how the flexibility and efficiency of Lua allow developers to rapidly prototype and iterate on game ideas, making it an ideal choice for interactive platforms like Roblox.

The history of the Lua programming language is both intriguing and significant, particularly within the realms of game development and software design. Created in 1993 by Roberto Lerusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes at the Computer Graphics Technology Group (TECGRAF) [1] of the Pontifical Catholic University of Rio de Janeiro, Lua emerged out of necessity during a period of trade barriers in Brazil. TECGRAF, faced with the inability to procure external software solutions, embarked on creating their own tools, ultimately giving birth to Lua as a lightweight, flexible language designed to meet the specific needs of their projects. Over time, Lua [2] evolved to become one of the most popular and versatile scripting languages, with particular prominence in game development.

This study explores the fascinating journey of Lua, from its origins to its current applications in platforms like Roblox, where it powers a vast range of games and interactive experiences. We will also delve into the language's influence on game development, focusing on the design and implementation of the Pao Tycoon game in Roblox. Through this exploration, we aim to highlight Lua's key features, its integration into modern development environments, and its pivotal role in shaping user experiences within dynamic, multiplayer gaming platforms.

II. FEATURES AND IMPLEMENTATIONS

2.1 Roblox Game Development with Lua

The popular gaming platform Roblox [3] first launched in 2006. Created two years earlier by David Baszucki and Eric Cassel, Roblox has become one of the world's leading interactive gaming environments. Roblox allows developers to utilize Roblox Studio, the specialized IDE for game programming, design and evaluation. The studio is free to use, and many young ones are encouraged to learn and become better programmers using it. The studio has a development, design, and testing space all within one environment, making the process of Roblox game development very efficient.

The primary language for Roblox scripting is Lua, sometimes referred to as Luau [4]. Lua is a high level, embeddable scripting language that supports multiple types of programming styles, including object-oriented, functional, data-driven, and several others [5]. Lua was designed to be used as an extension language that works embedded in a host program [6]. Lua is a standalone language, but it is also implemented as a library written in clean C, which makes it very compatible with the C-based languages [7]. C and C++ both have Lua libraries that can be imported and used without altering the syntax significantly. Lua takes qualities from several languages listed earlier in the study, but the language draws heavy similarities and influence from Python. A programmer with Python experience could learn Lua sufficiently in a short period of time.

With its beginner friendly syntax and efficient embeddable scripting, Lua is most commonly used as a scripting language for game development and design [8]. Some popular examples that utilize Lua are the video games Roblox, Garry's Mod, Angry Birds, World of Warcraft, and even the online banking app Venmo. Since Lua works well with C-based languages, Lua is typically reserved for object creation and game mechanics, while the memory handling is done with C/C++ in the background.

Roblox allows users from all ages to enjoy varying types of games created by developers and designers. There are several types of games that can be created in Roblox, with some of the more popular ones having several imitations designed after them. Some types are best described as simulation games, tycoon games, role-playing action games, and many more. There are over 40 million different Roblox games in circulation, and there are a few added every day.

As a lightweight, yet powerful scripting language, Lua is most notably known for its role in making Roblox code perform at efficient speeds and stable networking among countless players. Objects, mechanics, and other visuals are stored in the development part of the Roblox Studio in customizable file structures, seen in Fig. 1. While developers can add C-based code to customize memory handling, most of the memory management is handled in the background, and no knowledge of C/C++ is required to start developing [9].

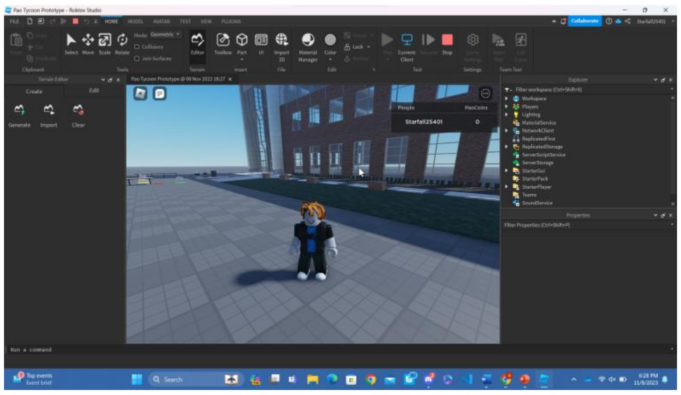

Figure 1. Development Window in the Roblox Studio Environment

The best way to learn game development with Lua is to create a game in the studio. Developing a game from start to finish is very straight-forward with Roblox Studio, so the fun part is deciding which type to use and what direction the game should go in. Should the game be an action-packed superhero game? A hidden identity murder mystery? A character role-playing costume party with players from around the world? The possibilities are endless with Roblox.

2.2 How Our Game and Other Games Work

For this projectwe decided to make Pao Tycoon, a tycoon style game. In this game, the player's goal is to collect PaoCoins, the choice of currency in the game. With PaoCoins, the player can upgrade their buildings and machines to collect even more PaoCoins. Fig. 2 shows one of the structures that the player can build with enough PaoCoins gathered. The game is intended to be interactive with multiple players, and the leaderboard shows the current ranking of players with their PaoCoin amounts. There is no winner in this style of game; instead, players are encouraged to keep collecting in the background, build more structures, and become wealthy with PaoCoins. If players obtain enough PaoCoins, players may even get to meet Pao!

This game mode style isn't too uncommon when it comes to the platform of Roblox. In fact, the idea of a tycoon is so widely used that there may be hundred if not thousands of players playing up to hundreds of different types of tycoons. The types of tycoons can range from a military base builder to an ice cream shop.

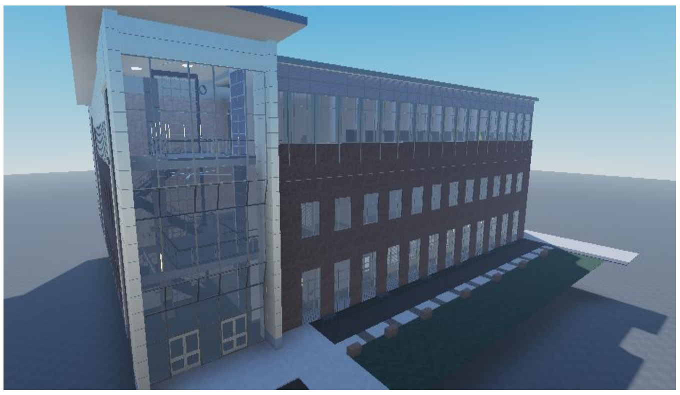

Figure 2. Recreation of the Computer Science and Technology Building at Southeastern Louisiana University in Roblox

How an ice cream shop tycoon may work can go multiple ways. However, most of the time it may go in the direction of selling ice cream in bundles and then using the earned currency to upgrade ingredients or create some sort of automation. The player may have to shove ingredients into a giant blender and then sell smaller shipments of ice cream from those used ingredients. Later down the line, the player might upgrade the size of the blender or even get two. Then, to increase the efficiency the player could get robots to do some work for them. The trend of automation and upgrading as users continue is the main premise of all tycoon games.

Now a Military Tycoon may be a little different. There will still be the components of upgrading over time, getting automation, and becoming more efficient to get more money. However, sometimes military tycoons can throw a little twist into things to make the game more alluring and active. Some military tycoon games are different by allowing players to raid other players' bases and collect more points that way. However, the other players can buy defenses with the points they acquired to make their base stronger and less vulnerable to raids. These defenses can range from AI shooting other players, walls, and traps. Players can even buy vehicles to attack other players or use them defensively to keep their points. The genre of Military tycoons can be significantly more hectic and is often on the more action based side than a normal tycoon usually is. There are hundreds of thousands of different types of tycoons that many players play, but those are the more common types on this platform.

2.3 Current Build of the Pao Tycoon

For this study, we have a basic example of what a Tycoon is. We have the point collection, a leaderboard, a path of upgrades, and ways to make the point collection more efficient. Most importantly, we have a beautiful building to house the main course of Pao's classroom. Within the classroom we have 56 computers that increase in point values as players continue along the table holding up the computers.

As time goes on, coins will fall onto the table and fall onto a despawner which will delete the coins and then add them to player's leaderboard score. The first computer spawns coins that are worth 1 point, then the next spawn a 5 point coin, and so on. From this point on, we can continue with multiple routes to get more points. For example, we can spawn vending machines, more computers, entire lab rooms, and much more. Similar to the way people fill out a building in real life, these methods are used to accumulate more points in the game. However, for the purpose of this study, we have reserved computers to show the basics of how tycoons and our code of choice, Lua, is used.

In Fig. 3 and Fig. 4, the world building capabilities of Roblox are showcased. Modeled after the Computer Science and Technology Building at our school, the objects in each part of the figure interact with each other to create a cohesive image. With these objects working together, we can create custom classrooms with tycoon elements in them.

2.4 What Else is Lua Used For?

Lua isn't just used for Roblox however. Lua can also be used to develop other games, web apps, and developer tools. For example, the mobile app Venmo was built using Lua. The video game Angry Birds was also made using Lua along with several game engines. These game engines can range from 2D to 3D. On the 2D side, the more popular ones would include: LOVE, Instead, Raylib, Gideros, Corona, and Defold. On the 3D side: GameGuru, Roblox Studio, Leadworks, Lumberyard, Shiva, Spring RTS, and Stingray.

A lot of popular games and gaming platforms support using Lua to extend, modify, or create new game systems. For example, in the massively multiplayer online role-playing game World of Warcraft, players can use Lua add-ons (mods) to customize the game interface used.

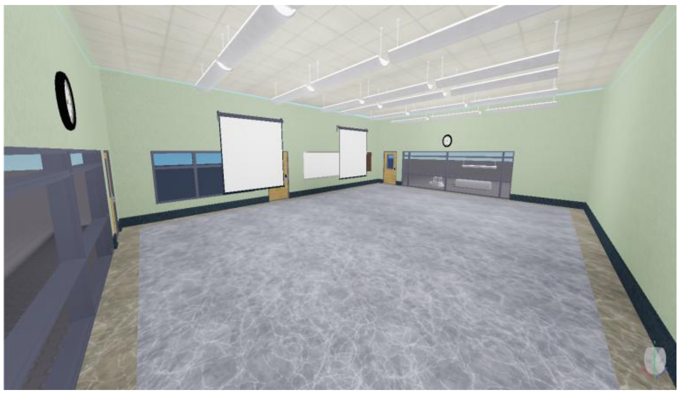

Figure 3. Roblox Model of a Typical Classroom in Fayard Hall at Southeastern Louisiana University in Pao Tycoon

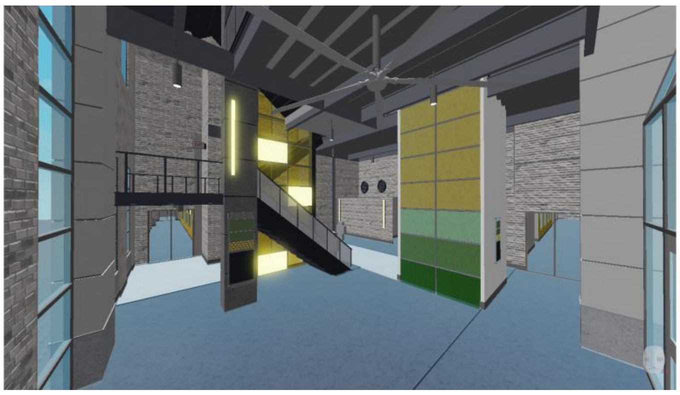

Figure 4. Roblox Model of the Computer Science and Technology Building Interior in Pao Tycoon

Something interesting to note is that Lua is one of, if not the most used scripting language used for modding published games. This is often the case because developers value the openness of using Lua to allow for flexibility for the community. Designers don't have to be a true game developer and know everything about the syntax to make a simple modification. Developers recognize that Lua is a very small light-wight programming language so it won't affect performance to a noticeable degree.

Lua can also be used with web development, data analysis, networking, software, and academic research. Programmers can take their skills one step further by using Lua to write networking scripts using Lua. A common networking tool known as Nmap is usually used to scan systems for open ports and vulnerabilities in online networks. As well as networking, Lua can be used to create ethical hacking tools for Nmap.

For Web development, Lua can be used to build web applications and developer tools, as well as being embedded in web servers, making it a popular choice for building high-performance applications. Lua has also been used in content management systems and e-commerce platforms. Some popular websites that are known to use Lua are Affirm, Venmo and Shopify. Affirm and Venmo are both known as finance related websites or services. Affirm is well known as a financial option working with Amazon that can allow monthly bills instead of paying for one bigger bill all at once. Venmo is more of a tool where friends and family can send money to each other for a more convenient experience. Shopify is known as an e-commerce platform that allows anyone to start, grow, manage, and scale businesses. It enables businesses to build an online store, market to customers, and accept payments across multiple channels and locations.

For Data Analysis, Lua was used particularly in the field of machine learning. It can be used to build models and algorithms for data analysis and prediction. One popular framework for machine learning that uses Lua is a framework known as Torch. Torch utilizes efficient linear algebra functions with GPU support, Neural Network packages with automatic differentiation, and multi-GPU support.

For Academic research, Lua has also been used to develop prototypes and proof-of-concept implementations of new algorithms and systems. One example is the development of domain-specific languages (DSLs).

2.5 Basic Building Blocks of Lua

Lua operates using fundamental building blocks known as data types. Like most programming languages, it includes familiar types such as integers, strings, characters, floats, booleans, and arrays. Different languages handle data types in different ways; for example, languages like Java and C/C++ require explicit type declarations so the compiler knows exactly what kind of data each variable holds. In contrast, languages such as Python and Lua use dynamic typing, meaning the programmer does not need to specify a type. Writing something like "x = 4" is sufficient, rather than declaring "int x = 4." There is one special data type used within Lua known as "nil". Nil is used as a data type to show the absence of a value returned from an expression. Fig. 5 shows a few usages of these types in the PaoCoins in Pao Tycoon.

```lua
local PaoCoin = script.Parent
local Players = game:GetService("Players")
local height = 0

PaoCoin.Touched:Connect(function (Object)
    if Object == game.Workspace.Despawner then
        PaoCoin:Destroy()
    end
    if Object.Parent:FindFirstChild("Humanoid") then
        PaoCoin:Destroy()
        local player = Players:GetPlayerFromCharacter(Object.Parent)
        local leaderstats = player.leaderstats
        local PaoCoinStat = leaderstats and leaderstats:FindFirstChild("PaoCoins")
        if PaoCoinStat then
            PaoCoinStat.Value += 1
        end
    end
end)

while true do
          PaoCoin.Orientation += Vector3.new(0,1,0)
          wait()
end
```

Figure 5. Lua Code for the PaoCoin Game Object

This study explores the profound connection between Lua's flexibility and the intricacies of our Roblox game design, delving into the unique aspects of Lua that propelled our project to success. Lua's lightweight, extensible nature emerged as a key factor in our development process, allowing for the creation of a rich and dynamic gaming experience. Lua's dynamic typing, coroutines, and metaprogramming capabilities empowered our development team to implement complex game mechanics with unparalleled ease. Lua's minimalist yet powerful syntax facilitated rapid prototyping and iteration, allowing us to adapt to continually growing design constraints.

Lua's integration within the Roblox platform emphasizes how Lua seamlessly interfaces with the game engine, providing an expressive and efficient scripting environment. Lua's unique attributes were harnessed to build a modular, scalable, and highly customizable architecture, elevating our Roblox game to a level of sophistication and interactivity that sets it apart in the gaming landscape.

The complete presentation version of Pao Tycoon in Roblox for the Computer Science and Technology Building is shown in Fig. 6. With all the building blocks of Lua, Pao Tycoon was built for the world to play, and players can enjoy gathering their PaoCoins and continue upgrading many buildings and structures in our school.

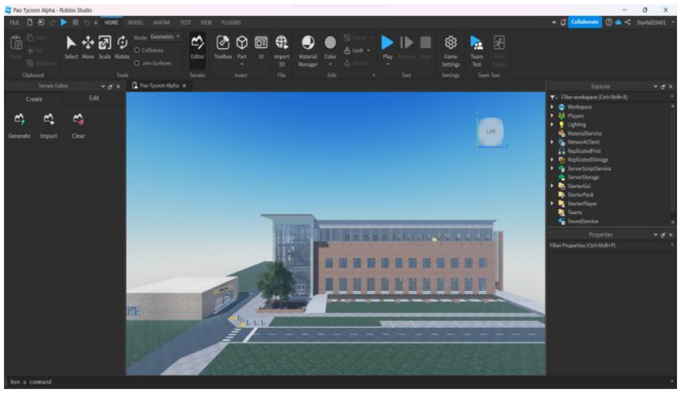
Figure 6. Complete Presentation Version of Pao Tycoon in Roblox

III. EVALUATION

After developing the Roblox game Pao Tycoon, Lua was clearly recognized as a high-level language with powerful applications in the game development and scripting scene. While many different games have been developed with the language, Pao Tycoon was designed with showcasing Lua’s efficient programming in mind. Several members can work on a project at the same time with minimal disruption, and results can be seen in real time for all members in Roblox Studio, and Roblox is such a well-known gaming platform that users can see the impact that it has on young developers and game designers.

In this project, we present a compelling case study on the adept utilization of the Lua programming language in the design and development of our groundbreaking Roblox tycoon. Lua's inherent flexibility, simplicity, and efficiency have empowered our development team to seamlessly integrate complex gameplay mechanics, dynamic environments, and interactive features, all while maintaining optimal performance.

The use of Lua scripting not only streamlined the development process but also facilitated quick iterations and adjustments, enabling us to respond rapidly to feedback and evolving design requirements. The report highlights how Lua's lightweight yet powerful nature played a pivotal role in enhancing the game's scripting capabilities, enabling intricate customization and fostering an engaging user experience. Readers will gain a profound appreciation for the pivotal role Lua played in the success of our Roblox game, showcasing its prowess as a programming language in the context of immersive and interactive gaming environments.

IV. CONCLUSION AND FUTURE WORK

The development of Pao Tycoon highlights how Lua’s simplicity, flexibility, and strong integration with Roblox Studio make it an effective language for creating interactive games. Through this project, we demonstrated how Lua’s lightweight design supports rapid prototyping, modular game mechanics, and smooth collaborative development. Beyond Roblox, Lua’s broad use in game engines, software tools, and research environments reflects its continued relevance as a versatile scripting language. Our case study shows that even with its minimalist structure, Lua provides the power needed to build dynamic and engaging gameplay experiences.

Future work includes expanding Pao Tycoon with additional buildings, upgraded systems, and more advanced multiplayer features, providing new opportunities to explore Lua’s capabilities in larger and more immersive environments. While the current game is limited to the Computer Science and Technology Building, future expansions may incorporate the rest of the campus, creating a richer and more engaging community experience.

REFERENCES

[1]. Lua, (2025). "The Evolution of an Extension Language: a History of Lua,"https://www.lua.org/history.html, November 2025.

[2]. Grumic, M., Vasic, M., Kovacevic, J., and Kastelan, I., (2015) "Porting of run-time environment for Lua-based applications," 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, Germany, pp. 128-131, September 2015,DOI: 10.1109/ICCE-Berlin.2015.7391213.

[3]. Roblox, (2025). "An Online Game Platform and Game Creation System,"https://www.roblox.com, November 2025.

[4]. Brumbaugh, Z., (2021)."Coding Roblox Games Made Easy: The ultimate guide to creating games with Roblox Studio and Lua programming," Packt Publishing.

[5]. Clark, D. L., (2009). "Powering intelligent instruments with Lua scripting," 2009 IEEE AUTOTESTCON, Anaheim, CA, USA, pp. 101-106, September2009, DOI: 10.1109/AUTEST.2009.5314042.

[6]. Ignjatov, N.,Bjelica, M. Z., Ćetković, M. and Radovanović, S., (2014), "Integration of Lua script interpreter for automatic device configuration using TR-069," 2014 22nd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, pp. 1118-1121, November 2014, DOI: 10.1109/TELFOR.2014.7034603.

[7]. Vukobrat, V.,Rankov, B.,Jovanović,P.,and Popović, M.,(2015). "Generating declarations needed by LuaJIT compiler for binding Lua and C," 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, pp. 847-850, November 2015,DOI: 10.1109/TELFOR.2015.7377598.
[8]. Geisler, B. J.,Mitropoulos, F. J.,and Kavage, S.,(2019). "GAMESPECT: Aspect Oriented Programming for a Video Game Engine using Meta-languages," 2019 SoutheastCon, Huntsville, AL, USA, pp. 1-8, April 2019,DOI: 10.1109/SoutheastCon42311.2019.9020369.
[9]. Berić, M., Bjelić, V., Simić, Đ., and Fimić, N.,(2017). "Implementation of user settings control with Lua C API," 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, pp. 411-414, October 2017,DOI: 10.1109/TELSKS.2017.8246312.