

Unity vs Unreal Engine: A Comparative Study

Urvi Rana

School of Computer Applications
MRIIRS, Faridabad
urviraana@gmail.com

Ayan Rai

School of Computer Applications
MRIIRS, Faridabad
ayanrai8287@gmail.com

Urvashi Sangwan

School of Computer Applications
MRIIRS, Faridabad
urvashibaswana.84@gmail.com

Abstract - Unity and Unreal Engine are leading forces in the game engine market. A significant gap exists in the literature for practical, direct comparisons that move beyond high-level feature lists. This study addresses that gap by presenting a controlled, hands-on evaluation. The methodology involves developing an identical game project on both platforms to provide a precise comparison of their performance, workflows, and resource management. Key aspects, including graphics rendering fidelity, mobile performance optimization, and programming language efficiency, are analyzed. Furthermore, their respective economic models are examined to provide a comprehensive perspective. This unique, parallel development approach yields concrete, comparative data, offering developers a robust framework for making informed engine selections.

Keywords : Unity, Unreal Engine, Game Development, Engine comparison, Graphics, Performance, Licensing

I. INTRODUCTION

The global game engine market is rapidly expanding, projected to reach 8.35 billion dollars by 2030 [13], with Unity and Unreal Engine as leading forces [10]. While most comparisons of these engines are theoretical, this research provides a novel and practical evaluation. This is achieved by developing an identical 3D cube-solving game on both platforms to provide a direct, hands-on comparison. This evaluation examines their performance, graphical capabilities, ease of use, and suitability for various project types, with a specific focus on providing actionable insights for developers. By analyzing both the technical and economic models, the goal is to equip developers with the concrete, comparative results needed to make informed engine selections based on real-world development experience.

A. Overview of Unity and Unreal Engine

To fully understand the difference between Unity and Unreal Engine, it's crucial to first understand their fundamental aspects. Unity is a cross-platform game engine created by Unity Technologies and it was launched in 2005 and often used for developing 2D and 3D games, simulations, virtual reality experiences, and other interactive applications. Due to its versatility, multi-platform compatibility, and user-friendly interface, it has become a popular choice for developers of all skill levels [14]. Unity's physics engine, graphics rendering capabilities, and animation system are among its primary characteristics, which allow for the development of complex and engaging gaming settings. Additionally, Unity's asset store simplifies the designing and development process by providing an extensive selection of ready made components, plugins, and themes [14]. On the other hand, Unreal Engine is a game engine developed by Epic Games and was first released in 1998. Initially, it

was exclusively employed for crafting high-performance, visually impressive games, such as the Unreal Tournament series. Over time, Unreal Engine transformed into a versatile tool capable of powering a diverse range of applications, including architectural visualizations, product demonstrations, and even films [8]. Its primary features includes robust physics engine, global illumination, and dynamic lighting, facilitating the creation of highly realistic and immersive environments [11]. Moreover, Unreal Engine's Blueprint visual scripting system allows developers to design complex game logic without extensive programming expertise [3].

B. This paper is organized as follows

Section II reviews existing literature comparing game engines, Section III details our methodology for side-by-side development, Section IV presents a technical comparison of both engines, Section V analyzes their economic models, Section VI compares their graphical capabilities with visual evidence, Section VII provides comparative data on market position and resource usage, Section VIII concludes our findings, and Section IX discusses practical recommendations for developers.

II. LITERATURE REVIEW

The selection of a gaming engine is a complicated decision influenced by a number of factors other than straightforward performance metrics, according to a comprehensive evaluation of the academic and commercial literature currently in availability. As TABLE I shows that while Unity may offer advantages in cross-platform development and accessibility for mobile and educational applications, Unreal Engine is often chosen for its superior high-fidelity rendering capabilities, according to the reviewed research.

TABLE I. COMPARATIVE ANALYSIS OF UNITY AND UNREAL ENGINE BASED ON LITERATURE REVIEW

Authors	Focus Area	Key Findings
Abramowicz et al. [2]	Performance Testing (Unity vs. Unreal)	Unity achieved better average frame rate while Unreal had superior high-fidelity rendering
Vuorinen et al. [5]	Animated Seq. Render.	Both engines have distinct strengths in animation rendering.
Pe'rez et al. [4]	Energy Consumption	Unity is more efficient on mobile, Unreal is optimized for high-end desktops.

DevClass[8]	Industry Preference	Unreal preferred (51%) over Unity (16%) in non-gaming media
Politowski et al. [10]	Engine Selection Factors	Choice depends: project, team size, expertise
Salama et al. [14]	Comparative Analysis	Unity: easy UI, cross-platform, large asset store; Unreal: high-end graphics, requires C++
Anuradha M. Sandi et al. [15]	Game Development with Unreal	Reviews tools, methods, UX, immersive design
Kuznetsov et al. [16]	Educational Use (Unity vs. XNA)	Unity is more accessible, promotes student-centered learning
Deshmukh et al. [1]	3D Game Project	Real-time rendering, immersive, comedic 3D delivery game
Lan Yuxi [6]	3D Modeling in Games	3D modeling blends art and tech, but still needs improvements

III. METHODOLOGY

This research employed a mixed-methods approach to provide a comprehensive comparison of Unity and Unreal Engine for game development. The methodology integrated practical game development with secondary data analysis of existing literature and market reports.

A. Practical Comparative Game Development

To gain firsthand insights into the development workflows, performance characteristics, and resource utilization of both engines, an identical 3D cube-solving game project was developed concurrently on both Unity and Unreal Engine.

This involved:

- **Identical Core Game Mechanics:** The fundamental gameplay loop, involving the rotation and manipulation of a 3D cube to achieve a solved state, was implemented with equivalent functionality in both engine environments.
- **Consistent Asset Utilization:** The same set of 3D cube models, textures, and basic lighting setups were imported and utilized within both engine projects to ensure a direct comparison of visual rendering and asset handling.
- **Performance Tracking:** Throughout the development and testing phases, key performance metrics were monitored using the built-in profiling tools of each engine.

The performance tracking included:

- **CPU Usage:** Measured to assess the processing overhead of the game logic and engine systems, as visualized in the provided graph showing low but consistent usage for an unspecified process.
- **Memory Usage:** Tracked to understand the memory footprint and management efficiency of each engine during gameplay, as visualized in the provided graphs showing a high, stable physical memory usage and a gradual increase in free memory.

- **Qualitative Visual Analysis:** A direct visual comparison of the rendered game output on identical hardware was conducted to subjectively assess the graphics quality achievable with similar lighting and basic material setups, noting Unreal Engine's demonstrably higher visual fidelity.
- **Retrospective Qualitative Workflow Assessment:** A retrospective reflection on the development workflow was conducted, focusing on the ease of use of the editor interface, the intuitiveness and efficiency of the scripting systems (C# in Unity, Blueprint and C++ in Unreal), the clarity of scene management, and the overall perceived efficiency of the development process within each engine.

The evaluation of the Retrospective Qualitative Process:

- **Unity:** The Unity editor offered an intuitive and well-structured interface, with logical placement of the Scene view, Hierarchy, Inspector, and Project browser. Scene manipulation through drag-and-drop was simple. The Inspector provided clear access to component properties. C# scripting was relatively easy to learn and use, benefiting from extensive online resources and a clear syntax for efficient coding. Unity's system for scene management, based on Game-Objects and Prefabs, was straightforward for organizing game elements and promoting reuse. This overall ease of use facilitated rapid prototyping.
- **Unreal Engine:** The Unreal Engine editor, though powerful, had a steeper learning curve. Its interface, including the Content Browser, World Outliner, Details panel, and Viewport, required more exploration. Scene setup and Actor manipulation felt less direct than Unity's GameObject system, despite the engine's visual strengths. Blueprint scripting offered a visual alternative but could become complex, while C++ provided performance and control but demanded more programming knowledge. Unreal Engine's scene management, using Levels and Actors, was robust for world building but required a more structured approach. Overall, development in Unreal Engine sometimes felt less immediately accessible and more deliberate than in Unity.

B. Secondary Data Analysis

Complementing the practical development, a thorough review of existing literature and industry reports was conducted to provide a broader context and validation for the practical findings. This involved analyzing the sources previously listed in the research paper.

C. Hardware and Software Specifications

The practical game development and testing were conducted on systems with the following specifications:

- RAM: 8GB
- OS: Windows 11 23H2
- PROCESSOR: 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz

IV. TECHNICAL COMPARISON

A. Graphics and Rendering

Both engines offer distinct advantages in visual capabilities as shown in Table II, Unreal Engine excels in rendering and physics [11], while Unity performs better on mobile platforms with less powerful hardware [7].

TABLE II. ENGINE PERFORMANCE HIGHLIGHTS

Engine	Performance Highlights
Unreal Engine	Unreal Engine demonstrates superior rendering engine performance, physics simulation, and world-building tools [11]
Unity	Unity demonstrates stronger performance on mobile platforms with more consistent frame rates on less powerful hardware [7].

B. Economic Analysis

According to Table III, Unity offers a free personal tier for developers earning under \$200,000 per year, with a price increase for their Pro and Enterprise tiers [12]. Unreal Engine is free to start and only charges a 5% royalty on revenue over \$1 million [9].

TABLE III. ENGINE PRICING DETAILS

Engine	Pricing Details
Unreal Engine	<ul style="list-style-type: none"> Free to start with 5% royalty on revenue over \$1 million No upfront costs for most features [9]
Unity	<ul style="list-style-type: none"> Free Personal tier for developers earning under \$200,000/year Pro subscription with 8% price increase Enterprise tier with 25% increase [12]

V. GRAPHIC COMPARISON

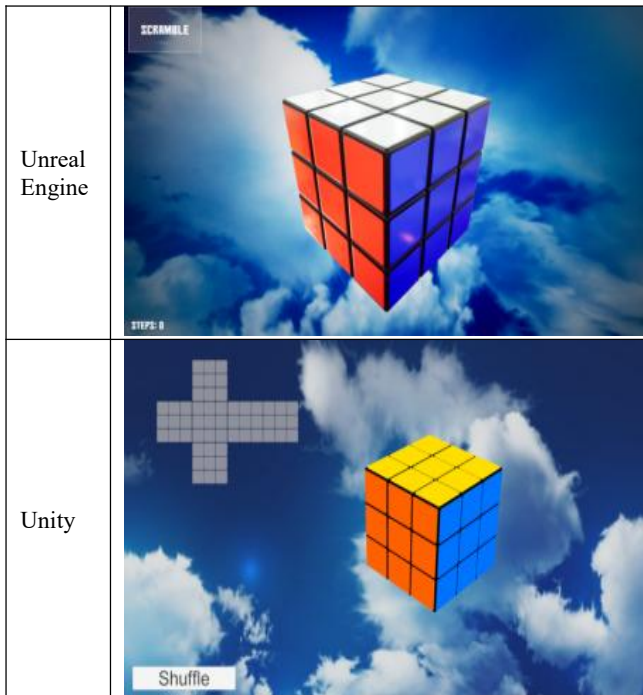


Fig. 1. Cube Solving Game

According to Fig.1, Unreal Engine shows high detail,

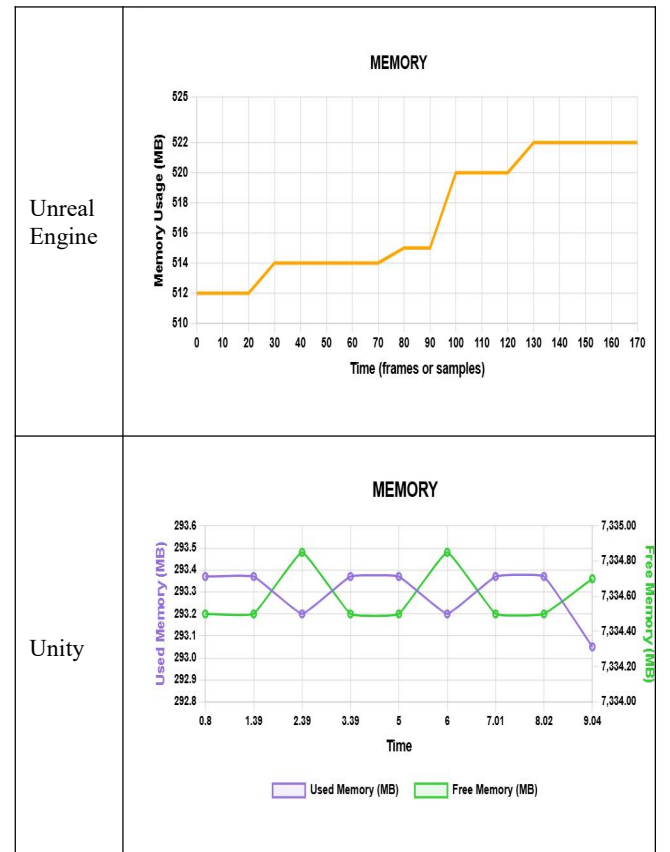
realistic lighting/shading and rich textures. Unity shows simpler graphics, flat shading, basic textures. Potential Advantages are Unreal Engine: Demonstrates a clear advantage in achieving high-fidelity visuals due to its robust rendering pipeline. This makes it well-suited for projects prioritizing visual realism and immersion, as supported by the literature on its superior graphics capabilities. Unity: Demonstrates performance and ease of development for simpler visual requirements. By utilizing less complex rendering, it can potentially achieve better frame rates, especially on less powerful hardware.

VI. COMPARATIVE TABLES

A. Memory Usage

According to TABLE IV, Unreal Engine maintains stability and predictability while using additional memory i.e. it jumps from 1000MB to 2000MB. Unity consumes less memory (750-850MB), but it fluctuates more frequently as it cleans up unwanted data. Unreal is better for big games that need lots of memory upfront, while Unity is better for smaller games that want to save memory. Unity utilizes around half as much memory as Unreal Engine.

TABLE IV. COMPARISON CHART BASED ON MEMORY USAGE

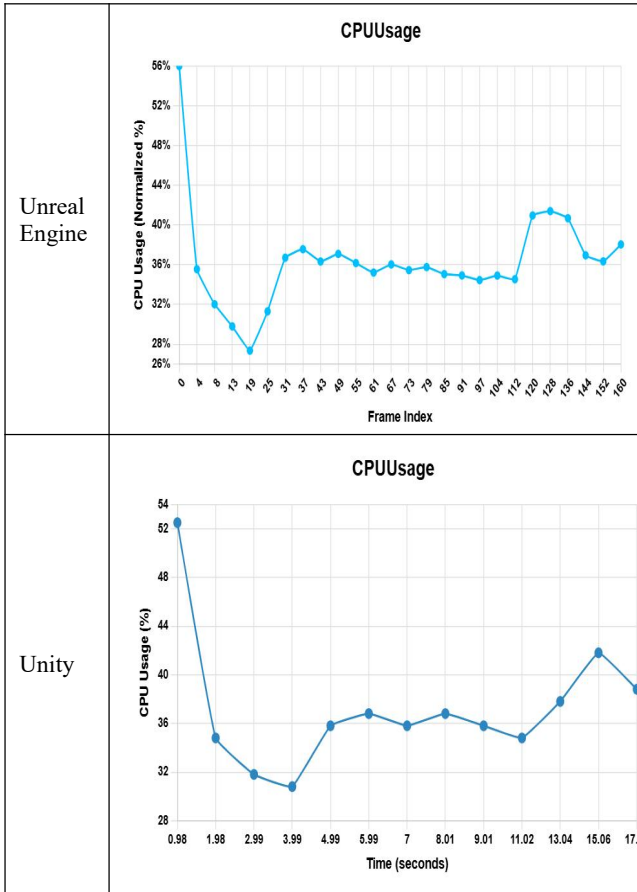


B. CPU Usage

According to TABLE V, the CPU behavior displayed by both engines appears similar. Both starts high at 50% and then rapidly decline until settling at 30% to 35%. Throughout the test period, Unity displays more frequent

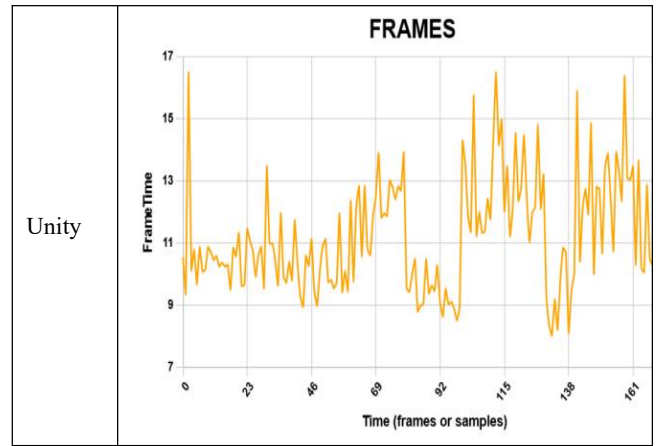
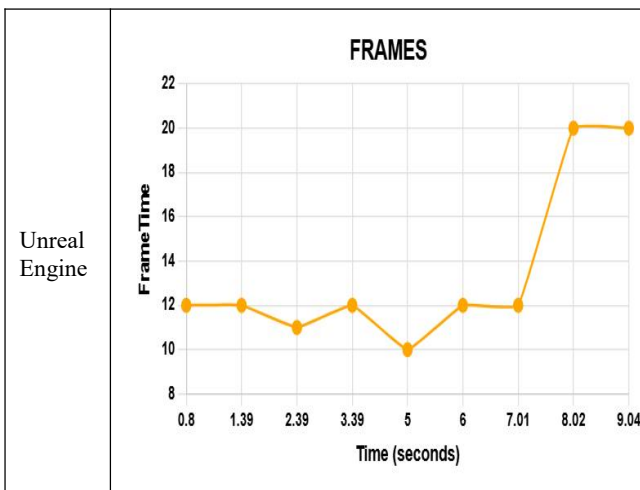
minor swings in CPU utilization, whereas Unreal Engine maintains a little more consistent CPU usage with fewer spikes. After initial startup, the CPU burden is effectively reduced by both engines, although Unreal offers more reliable and constant performance. Unreal's CPU utilization stabilizes into a more consistent pattern, whereas Unity's fluctuates more. Overall, Unreal has a small advantage in stability but both engines perform in terms of CPU economy.

TABLE V. COMPARISON CHART BASED ON CPU USAGE



C. Frame Time

TABLE VI. COMPARISON CHART BASED ON FRAMES PER SECOND



According to TABLE VI, With steady 9-10 ms performance and a fluid 60+ FPS throughout testing, Unity exhibits exceptional frame time consistency. Unreal Engine shows significant variation between 8 and 16 ms, with observable spikes that could cause stuttering. Even if both engines maintain respectable average performance below the 16.7ms mark, Unity is the better choice for smooth, consistent frame delivery because of its higher stability.

VII. CONCLUSION

The side-by-side development of an identical game on both platforms allowed for a concrete, real-world comparison of Unity and Unreal Engine. The primary gain from this unique methodology is a clear understanding of the practical trade-offs between the two engines. The findings from this research reveal that Unreal Engine excels in visual quality and complex simulations, making it the ideal choice for projects that prioritize photo-realistic graphics and require advanced technical skills. In contrast, Unity's efficient development workflow, quicker iteration cycles, and user-friendly interface prove it to be the better option for rapid prototyping and for smaller teams operating with budget limitations. Ultimately, this study demonstrates that neither engine is a universal solution, but rather a specialized tool best suited for distinct development priorities.

A. Future Work

- Larger-Scale Projects: Using this side-by-side approach to complex games with AI systems, multiplayer capabilities, and large environments.
- Long-Term Resource Profiling: Assessing memory and CPU utilization in greater detail during prolonged gaming sessions in order to determine scalability and optimization requirements.
- Platform-Specific Deployment: In order to assess performance, compatibility, and user experience on each platform, both engines will be evaluated across mobile, virtual reality, and console targets.

VIII. DISCUSSION

This study compared Unity and Unreal Engine by creating an identical 3D cube-solving game on each platform. The results highlights the following practical differences between the two engines :

- Graphics Rendering: Unreal Engine outperformed

Unity in terms of visual quality, displaying superior rendering and lighting with the same assets and setup.

- **Development Workflow:** Unity's user-friendly editor, C# scripting, and quick iteration cycles, provided a more efficient workflow for rapid prototyping, especially for smaller teams.
- **Performance and Resource Usage:** Both engines performed consistently, although monitoring revealed the distinct memory and CPU utilization patterns. Unreal's higher graphical fidelity comes with more complex resource management requirements.
- **Cost Structure:** Unity's free-to-use model for small teams lowers the entry barrier but becomes more expensive at scale. Unreal's royalty-based approach is cost-effective initially but scales with project revenue.

A. Recommendation

Unreal Engine for projects that:

- Prioritize cutting-edge graphics and photorealistic visuals.
- Have a team with advanced technical skills and experience in C++.
- Involve complex physics simulation or large, open-world environments.
- Target high-end platforms like PC and console.
- Benefit from the Blueprint visual scripting system for non-coders and rapid prototyping.
- Focus on non-gaming applications (e.g., virtual production, film, industrial simulations) where visual fidelity is critical.

Unity for projects that:

- Need rapid development, quick iteration, and an easier learning curve.
- Prioritize ease of use for smaller teams or teams with less programming experience.
- Demand faster build times and smoother testing cycles.
- Take advantage of Unity's extensive asset store and plugin ecosystem.
- Focus on mobile games, 2D games, or educational applications.
- Are constrained by budget and prefer royalty-free publishing models for small to mid-sized teams.

IX. REFERENCES

1. Deshmukh, Prof Tripathi, Adarsh Palkar, Ayush Sali, Shantanu. (2025). 3D Game Development. International Journal of Advanced Research in Science, Communication and Technology. 294-305.
2. Abramowicz, K., Borczuk, P. (2024). Comparative analysis of the performance of Unity and Unreal Engine game engines in 3D games. Journal of Computer Sciences Institute.
3. Gintere, I., Rovithis, E., Bakk, A' .K. (2024). Imgame project: A comprehensive theory of immersive aesthetics and innovation in serious gaming. International Journal of Gaming and Computer-Mediated Simulations.
4. .Pe'rez, C., Vero'n, J., Pe'rez, F., Moraga, M., Calero, C. (2024). A comparative analysis of energy consumption between the widespread unreal and unity video game engines. arXiv preprint.
5. Vuorinen, T. (2024). Animated sequence rendering performance comparison: Unity and Unreal Engine. Theseus.
6. Lan Yuxi (2023). Development and Application of 3D Modeling in Game. Academic Journal of Science and Technology. 7. 94-97.
7. Ciekanowska, A., Kiszczak-Glin'ski, A. (2021). Comparative analysis of Unity and Unreal Engine efficiency in creating virtual exhibitions of 3D scanned models. Journal of Computer Sciences Institute.
8. Survey finds game engines used equally by non-game projects, rise of interest in open-source Godot. <https://devclass.com/2024/08/23/survey-finds-game-engines-used-equally-by-non-game-projects-rise-of-interest-in-open-source-godot/>
9. Unreal Engine 5.5 Documentation. <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-5-documentation>
10. Politowski, C., Petrillo, F., Montandon, J.E. (2021). Are game engines software frameworks? A three-perspective study. Journal of Systems and Software.
11. S'm'id, A. (2017). Comparison of Unity and Unreal Engine. Czech Technical University in Prague.
12. Unity Pricing Updates. <https://unity.com/products/pricing-updateskey-updates>
13. Game Engines Market Analysis Report. <https://finance.yahoo.com/news/game-engines-market-analysis-report-124800759.html>
14. Salama, R. Elsayed, M. (2021). A live comparison between Unity and Unreal game engines. Global Journal of Information Technology: Emerging Technologies. 11(1), 001-007.
15. Anuradha M. Sandi*, Abdul Qadeer Makki, Krutika S, Mahadevi (1) 3D GAME DEVELOPMENT USING UNREAL ENGINE. International Journal Of Innovation In Engineering Research Management UGC APPROVED NO. 48708, EFI 5.89, WORLD SCINTIFIC IF 6.33, 11(8),164-169.
16. Kuznetsov, Vladyslav Moiseienko, Mykhailo Moiseenko, Natalie Rostalny, Bohdan Kiv, Arik. (2020). Using Unity to Teach Game Development.