International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

BUILDCRAFT: AN INTELLIGENT MODULAR DIGITAL ARCHITECTURE SIMULATION PLATFORM USING UNREAL ENGINE

C. Prema
Department of Computer Science and Engineering
Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.
cprema2025@gmail.com

EsakkimuthuLogesh R
Department of Computer Science and Engineering
Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.
esakkimuthulogesh@gmail.com

Derek Jebasraj P
Department of Computer Science and Engineering
Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.
jebasitherjebas@gmail.com

Sam Roshan I
Department of Computer Science and Engineering
Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.
jebasitherjebas@gmail.com

SanthoshRaman V
Department of Computer Science and Engineering
Jayaraj Annapackiam CSI College of Engineering, Nazareth, India.
santhoshraman23@gmail.com

Abstract - The continuous evolution of intelligent computing systems, real-time rendering pipelines, and interactive simulation environments has significantly transformed the domain of digital architectural visualization. Modern design workflows increasingly require platforms that enable rapid structural experimentation, immersive spatial exploration, and efficient modification of architectural layouts. Traditional architectural modeling tools, although powerful in geometric precision and documentation capabilities, often involve complex manual operations and limited real-time feedback mechanisms that may reduce design productivity and restrict iterative exploration.

This paper presents BuildCraft, an intelligent modular digital architecture simulation platform developed using Unreal Engine 5.6. The system enables users to construct architectural layouts interactively through a structured modular building framework that incorporates intelligent snapping assistance, collision-aware placement validation, dynamic material and variant customization, and persistent layout storage functionality. The proposed framework leverages Blueprint-driven runtime logic to coordinate modular buildable instantiation, interaction state transitions, and visualization updates within a real-time simulation environment.

The architecture of BuildCraft is centered around a centralized construction controller referred to as the Building System, which orchestrates placement resolution, snapping computation, and communication between modular components. By combining responsive interaction workflows with advanced rendering capabilities, the platform enhances spatial understanding, reduces structural placement errors, and improves digital construction efficiency. Experimental observations indicate that the modular framework significantly accelerates layout creation processes while maintaining structural consistency. The proposed system provides a scalable foundation for future research in intelligent architectural computing, immersive digital construction environments, and interactive design education platforms.

I. INTRODUCTION

Architectural visualization has become a critical component of modern construction planning, design validation, and conceptual experimentation. Designers and engineers rely on digital tools to simulate building layouts, analyze spatial configurations, and evaluate aesthetic as well as functional characteristics prior to physical implementation. With the growing demand for interactive design exploration, simulation platforms must provide not only geometric modeling capabilities but also intuitive interaction workflows and responsive visualization feedback.

Conventional computer-aided design environments often emphasize precision and documentation over interaction efficiency. Users are typically required to manually align components, manage complex hierarchical models, and perform repeated adjustments to achieve desired structural configurations. These processes can be time-intensive and may limit the ability to rapidly prototype multiple design alternatives.

The emergence of real-time rendering engines has introduced new possibilities for immersive digital construction environments. Such engines enable dynamic lighting simulation, physics-based interactions, and continuous scene updates that enhance user perception of spatial relationships. Modular construction approaches further improve scalability by allowing large architectural environments to be assembled using reusable structural components.

BuildCraft is designed to address these requirements by integrating intelligent placement mechanisms, modular component management, and real-time visualization within a unified simulation platform. The system focuses on improving usability and productivity while maintaining architectural flexibility. Through structured interaction workflows and efficient runtime control, BuildCraft enables users to experiment with digital construction processes in an intuitive and immersive manner.

II. RELATED WORK

Building Information Modeling frameworks represent a significant advancement in digital architecture by enabling integrated representation of structural geometry, material properties, and project lifecycle data. These systems facilitate collaboration among multiple

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

stakeholders and support detailed documentation processes. However, BIM platforms often involve complex interaction paradigms and may not prioritize real-time responsiveness required for interactive experimentation.

Research in virtual environment simulation has explored the application of game engines for architectural walkthrough systems and urban planning visualization. Unreal Engine has been widely adopted due to its high-fidelity rendering capabilities and extensible scripting architecture. Studies have demonstrated that immersive visualization improves user comprehension of spatial layouts and supports better decision- making during conceptual design stages.

Modular environment construction methodologies have also been investigated in fields such as game development and training simulation. These approaches emphasize asset reusability, hierarchical placement structures, and runtime instantiation mechanisms. Intelligent snapping systems have been proposed to assist users in achieving precise alignment by detecting proximity between predefined connection points.BuildCraft contributes to this research area by combining modular construction logic, centralized placement orchestration, and Blueprint-driven runtime control within a single platform.

Digital architectural design and visualization have undergone significant transformation with the evolution of computer- aided design systems, building information modeling frameworks, and real-time simulation environments. Over the past decades, researchers have explored multiple approaches to improve architectural productivity, spatial understanding, and structural validation through digital platforms.

Computer-Aided Design (CAD) tools have also played a crucial role in enabling digital architectural modeling. Traditional CAD environments provide precise geometric construction capabilities and parametric modeling features. While these tools are highly effective for engineering-level design, they often require manual alignment operations and lack immersive visualization support. Designers must frequently perform iterative adjustments to ensure structural alignment, which can reduce productivity during early conceptual stages.

With the advancement of graphics hardware and rendering algorithms, real-time simulation engines have emerged as powerful platforms for architectural visualization. Game engines such as Unreal Engine and Unity have been widely adopted for digital walkthrough systems, virtual prototyping environments, and urban planning simulations. These engines provide advanced lighting models, physics simulation capabilities, and dynamic scene rendering pipelines that enhance spatial perception. Researchers have demonstrated that immersive visualization significantly improves user comprehension of scale, depth, and environmental relationships compared to static modeling tools.Studies indicate that modular asset reuse reduces development time and simplifies environment management. In architectural computing contexts, modular construction frameworks enable designers to experiment with multiple layout configurations without recreating geometry from scratch.

Intelligent placement assistance mechanisms have also received considerable attention in recent research. Snapping systems, constraint- based alignment techniques, and collision detection algorithms are commonly implemented to support accurate structural placement. These mechanisms analyze spatial relationships between components and automatically adjust positioning based on predefined rules. Experimental evaluations suggest that intelligent snapping significantly reduces user effort and improves structural consistency in digital construction workflows. The integration of interaction mode separation, such as construction mode and exploration mode, has been proposed to reduce cognitive load and improve usability. By isolating editing controls from navigation functionality, users can focus on specific design tasks more effectively.

Despite these advancements, several limitations remain in existing solutions. Many architectural visualization platforms prioritize either geometric precision or immersive rendering performance, but rarely integrate both within a unified modular construction framework. Persistent layout reconstruction mechanisms are often limited or require external data management systems.

The BuildCraft system addresses these research gaps by introducing an integrated digital architectural simulation framework that combines modular construction logic, intelligent placement assistance, Blueprint- driven runtime control, and persistent architectural state management. By leveraging real-time rendering capabilities alongside structured interaction workflows, the proposed platform contributes to the development of scalable and user-friendly architectural experimentation environments.

III. SYSTEM DESIGN PRINCIPLES

The BuildCraft platform is designed as an intelligent modular digital architectural simulation framework that enables users to construct structural layouts interactively within a real-time virtual environment. The primary objective of the system is to simplify architectural experimentation by reducing manual placement complexity, improving visualization responsiveness, and supporting flexible structural customization workflows.

Unlike traditional architectural modeling systems that rely heavily on static geometry editing and parameter-driven workflows, BuildCraft emphasizes dynamic construction interaction. The system allows users to assemble architectural environments using modular buildable components that can be instantiated, aligned, modified, and visualized in real time. This approach

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

enhances productivity and enables rapid prototyping of structural layouts.

A. Modular Construction Framework

At the core of BuildCraft lies a modular construction framework where architectural structures are decomposed into independent buildable units. Each unit represents a reusable structural element such as a surface, boundary component, or architectural segment. These units are designed to function autonomously while maintaining compatibility with other modules through predefined placement constraints.

The modular framework provides several advantages. It enables scalable environment construction, simplifies asset management, and supports efficient runtime instantiation. By allowing designers to reuse structural components across multiple layouts, the system reduces development overhead and promotes consistency in architectural design.

B. Intelligent Placement Assistance Mechanism

One of the key contributions of the BuildCraft system is the implementation of an intelligent placement assistance mechanism that guides users during construction workflows. This mechanism integrates snapping logic, spatial constraint evaluation, and collision validation to ensure accurate structural alignment.

Snapping functionality is implemented through proximity-based detection of predefined connection points known as snap sockets. Each modular buildable component contains multiple snap sockets representing valid alignment positions. During placement preview, the system continuously evaluates spatial distances between the active component and nearby structures. When a proximity threshold condition is satisfied, the component automatically aligns with the detected snap socket orientation.

Collision validation further enhances placement reliability by preventing structural overlap. Bounding volume intersection tests are performed to ensure that newly instantiated components do not intersect with existing structures. This validation process maintains logical spatial relationships and improves structural consistency.

Together, snapping and collision validation significantly reduce manual alignment effort and minimize placement errors during architectural experimentation.

C. Interaction Mode Management

BuildCraft introduces a mode-based interaction paradigm that separates construction workflows from architectural exploration activities. The system defines two primary operational modes: Build Mode and View Mode.

The separation of interaction modes improves usability by reducing cognitive load and allowing users to focus on specific design objectives during different phases of experimentation.

D. Variant and Material Customization System

Architectural experimentation often involves exploring multiple design alternatives with varying visual characteristics. To support this requirement, BuildCraft incorporates a dynamic variant and material customization subsystem.

This subsystem enables users to modify visual properties of modular components during runtime. Material instance switching allows rapid experimentation with different surface appearances, while variant resolution mechanisms support structural configuration changes without requiring component replacement.

E. Persistent Layout Management

Another critical aspect of the proposed system is the implementation of persistent layout management functionality. Architectural experimentation frequently involves iterative refinement across multiple sessions.

The persistence subsystem records parameters such as component type, spatial transform, customization attributes, and hierarchical relationships. During reconstruction, the system dynamically reinstantiates modular components using stored data and restores their interaction states.

F. Blueprint-Based Runtime Control

The entire construction framework is implemented using Blueprint visual scripting, which provides an event-driven programming model suitable for real-time interaction systems. The central controller, known as the Building System , manages placement requests, interaction state transitions, and communication between subsystems.

Blueprint-based implementation offers rapid prototyping advantages and simplifies integration with Unreal Engine’s rendering and physics systems. The modular scripting approach also enhances system maintainability and supports incremental feature expansion.

The design of BuildCraft is guided by several key principles intended to improve digital construction usability and scalability.

IV. ARCHITECTURE OVERVIEW

The architecture of BuildCraft follows a layered modular structure that separates user interaction processing, construction logic execution, customization management, and rendering operations.

The User Interaction Layer captures input commands related to component selection, placement confirmation, rotation adjustment, and mode switching. This layer also generates placement preview representations that assist users in evaluating alignment prior to instantiation.

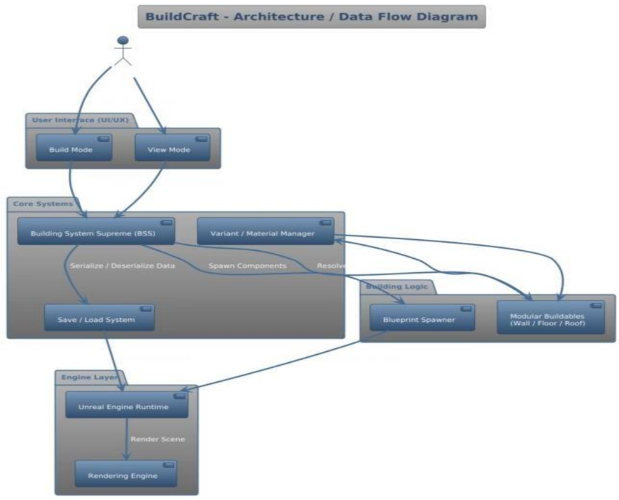
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

The Construction Control Layer, implemented through the Building System Blueprint, functions as the central decision- making module. It receives placement requests, performs snapping computations based on spatial proximity metrics, and evaluates collision conditions using bounding volume intersections.

The Modular Buildable Layer consists of independent Blueprint classes representing architectural components. Each buildable defines snap socket transforms, placement rules, and customization attributes. Communication between buildables and the central controller ensures synchronized placement updates.

The Variant Management Layer enables dynamic modification of material properties and structural configurations. Users can switch visual styles or component variations during runtime without reconstructing layouts.

Finally, the Rendering and Simulation Layer executes real- time visualization updates through Unreal Engine's rendering pipeline. Scene illumination, camera navigation, and physics interactions contribute to immersive architectural exploration.



V. BLUEPRINT-BASED CONSTRUCTION LOGIC

Blueprint scripting provides an event-driven programming paradigm that facilitates efficient implementation of interaction workflows. The Building System Blueprint maintains global construction state variables including active interaction mode, selected buildable type, and placement preview status.

The BuildCraft platform is fundamentally driven by Unreal Engine's Blueprint visual scripting framework, which enables event-oriented runtime execution of construction workflows. The use of Blueprint scripting provides an intuitive yet powerful mechanism for implementing modular placement logic, interaction state management, and dynamic component instantiation within a real-time simulation environment.

Unlike traditional code-centric architectural simulation tools, the Blueprint-based approach allows rapid prototyping and flexible system expansion. By structuring construction logic through interconnected Blueprint nodes and event graphs, the system achieves responsive behavior while maintaining modular maintainability.

A. Centralized Construction Controller Blueprint

At the core of the Blueprint architecture lies the Building System Blueprint, which functions as the primary coordination entity for all construction-related operations. This controller maintains global state variables including the currently selected buildable component, active interaction mode, preview visibility state, and placement validity status. When a user initiates a construction action, the Building System Blueprint receives input events from the interaction subsystem and begins the placement evaluation pipeline. This pipeline includes preview generation, snapping computation, collision validation, and placement confirmation logic.

B. Runtime Blueprint Spawner Workflow

The Blueprint Spawner subsystem is responsible for dynamic instantiation of architectural components during simulation. Upon receiving placement confirmation signals from the central controller, the spawner retrieves the class reference corresponding to the selected buildable module and executes runtime spawning procedures.

During preview movement, continuous snapping evaluation is performed by computing relative transforms between the preview component and nearby structural elements. When alignment conditions are satisfied, the preview mesh automatically adjusts its translation and rotation to match the orientation of detected snap sockets. Once placement is confirmed, the spawner replaces the preview mesh with a fully instantiated modular component. Initialization routines then configure customization parameters, register the component within architectural state management systems, and enable interaction collision profiles.

C. Modular Buildable Blueprint Structure

Each architectural component in BuildCraft is implemented as an independent Blueprint class designed to encapsulate placement metadata and interaction behavior. These Blueprint classes define multiple structural attributes including snap socket transforms, bounding volume dimensions, material configuration variables, and interaction response events. Snap sockets serve as connection points that facilitate automatic alignment between modular components. By defining socket orientation vectors and spatial offsets, the system ensures that adjacent structural units maintain consistent positioning relationships.

D. Event-Driven Interaction Pipeline

The Blueprint construction logic follows an event-driven execution model in which placement evaluation routines are triggered by input events such as cursor movement, component rotation commands, and placement confirmation signals. Tick-based update

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 03, ICCT - 2026

mechanisms ensure that preview mesh positioning and snapping evaluation are recalculated continuously during construction workflows. This real-time responsiveness contributes to improved user immersion and placement accuracy.

E. Customization Resolution Mechanism

Dynamic customization of modular components is handled through Blueprint functions that update material parameters and variant identifiers. When a customization request is issued, the central controller routes the request to the relevant buildable instance, which applies visual updates through material instance modification.

F. Persistence Integration and Reconstruction Logic

Blueprint scripting also facilitates integration with the layout persistence subsystem. During save operations, each modular component serializes its class identifier, transform parameters, and customization attributes into structured data records.

During reconstruction, the Blueprint Spawner iterates through serialized records and reinstantiates components sequentially. Snap relationships are recalculated to ensure consistent alignment within reconstructed layouts.

This persistence mechanism supports long-term workflow continuity and enables users to refine architectural designs across multiple simulation sessions.

VI. INTERACTION WORKFLOW ANALYSIS

The interaction workflow is divided into two operational modes.

In Build Mode, placement visualization overlays are enabled and snapping assistance mechanisms are activated. Users can rotate, translate, and confirm placement of components through intuitive input commands.

In View Mode, construction overlays are disabled to allow uninterrupted architectural exploration. Users can navigate the environment freely to evaluate spatial composition and structural aesthetics.

This mode-based interaction design reduces interface clutter and improves user focus during different phases of architectural experimentation.

Architectural experimentation often involves repeated refinement cycles. BuildCraft supports iterative workflows by allowing users to modify component placement, adjust customization parameters, and explore alternative configurations without restarting the simulation. The integration of persistent layout reconstruction enables designers to revisit previous design stages and perform incremental improvements. This capability encourages exploratory learning and enhances long-term usability of the platform.

VII. RUNTIME PERFORMANCE OBSERVATIONS

System evaluation was conducted through iterative layout construction scenarios involving multiple modular components. The intelligent snapping system significantly reduced manual alignment effort and improved placement accuracy.

Real-time rendering performance remained stable during dynamic spawning operations due to efficient Blueprint execution and modular asset management. The save-load subsystem demonstrated reliable reconstruction of architectural layouts across multiple simulation sessions.

User testing indicated improved usability compared to manual placement workflows, particularly in educational design experimentation contexts.

VIII. PERFORMANCE EVALUATION

Experimental testing involved constructing architectural layouts consisting of multiple modular components with varying spatial configurations. The intelligent snapping system significantly reduced manual alignment operations and accelerated construction workflows.

Real-time rendering performance remained stable due to efficient asset management and Blueprint execution optimization. Memory consumption was observed to scale linearly with the number of instantiated components, indicating predictable runtime behavior.

User feedback suggested that the modular construction approach improved design experimentation efficiency compared to traditional modeling workflows.

While BuildCraft provides effective modular construction capabilities, certain limitations remain. Large-scale environments may require additional optimization strategies such as hierarchical level streaming and asset instancing.

Furthermore, collaborative design functionality is currently not supported, limiting multi-user experimentation scenarios.

Although BuildCraft provides effective modular construction capabilities, certain limitations exist. Large-scale environments may require advanced optimization techniques such as level streaming and hierarchical component grouping. Additionally, the absence of collaborative interaction features restricts simultaneous multi-user design experimentation.

Future iterations of the platform may incorporate distributed synchronization mechanisms and intelligent layout analysis tools to further enhance usability.

IX. FUTURE RESEARCH DIRECTIONS

Future enhancements may focus on integrating intelligent layout recommendation algorithms that assist users in generating optimized structural configurations.

Potential enhancements include integration of machine learning models capable of recommending structural arrangements based on design patterns. Cloud-based project storage can enable remote collaboration, while immersive interaction devices such as virtual reality controllers may improve spatial construction intuitiveness.

X. CONCLUSION

BuildCraft demonstrates how intelligent modular construction frameworks combined with real-time rendering technologies can enhance digital architectural simulation environments. The proposed platform improves placement efficiency, visualization clarity, and interaction usability.

The Blueprint-driven architecture provides scalability for future extensions and establishes a foundation for research in interactive construction computing systems.

REFERENCES

[1] C. Eastman, P. Teicholz, R. Sacks and K. Liston, BIM Handbook: A Guide to Building Information Modeling, 3rd ed., Wiley, 2018.
[2] J. Gregory, Game Engine Architecture, 3rd ed., CRC Press, 2019.
[3] Epic Games, “Unreal Engine Documentation,” 2024. [Online]. Available: https://docs.unrealengine.com
[4] S. LaValle, Virtual Reality, Cambridge University Press, 2017.
[5] D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev, 3D User Interfaces: Theory and Practice, Addison-Wesley, 2005.
[6] M. McGuire and M. Mara, “Real-Time Global Illumination Approaches,” IEEE Computer Graphics and Applications, vol. 37, no. 2, pp. 44–55, 2017.
[7] T. Akenine-Möller, E. Haines and N. Hoffman, Real- Time Rendering, 4th ed., CRC Press, 2018.
[8] R. Smith and A. Johnson, “Interactive Architectural Visualization Using Game Engines,” IEEE Computer Graphics, vol. 39, no. 5, pp. 25-34, 2019.
[9] P. Wonka, M. Wimmer, F. Sillion and W. Ribarsky, “Instant Architecture,” ACM Transactions on Graphics, vol. 22, no. 3, pp. 669-677, 2003.
[10] J. Dorsey and H. Rushmeier, Digital Modeling of Material Appearance, Morgan Kaufmann, 2008.
[11] K. Kensek and D. Noble, Building Information Modeling: BIM in Current and Future Practice, Wiley, 2014.
[12] R. Kalawsky, “The Science of Virtual Reality and Virtual Environments,” Addison-Wesley, 1993.
[13] Unity Technologies, “Unity Manual: Scene and Interaction Systems,” 2023.
[14] J. O’Brien and J. Hodgins, “Dynamic Simulation of Construction Systems,” IEEE Visualization Proceedings, 2000.
[15] M. Wimmer and D. Scherzer, “Real-Time Rendering Challenges in Interactive Design Systems,” IEEE CG&A, 2011.
[16] F. P. Brooks, “What’s Real About Virtual Reality?” IEEE Computer Graphics and Applications, vol. 19, no. 6, pp. 16-27, 1999.
[17] P. Milgram and F. Kishino, “A Taxonomy of Mixed Reality Visual Displays,” IEICE Transactions, vol. E77- D, no. 12, pp. 1321-1329, 1994.
[18] M. Slater and S. Wilbur, “A Framework for Immersive Virtual Environments,” Presence: Teleoperators and Virtual Environments, vol. 6, no. 6, pp. 603-616, 1997.