iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72530

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Gamified Video Conferencing for Young Learners: System and Evaluation

Amaan Alam[1], Nitin[2], Shivaay Bathla[3], Yash Sharma[4]
Noida Institute of Engineering and Technology, Greater Noida, AKTU

Abstract: We present a gamified video conferencing platform that integrates real-time WebRTC-based video chat with multiplayer educational games (using Phaser3 and Colyseus) to boost engagement in children (ages 5–12). The system uses a web front-end (React/JavaScript) for the UI, PeerJS/WebRTC for peer-to-peer video, and a Node.js server (with Colyseus) to synchronize game state. In a pilot evaluation with 30 young learners, the gamified platform produced markedly higher engagement and learning gains than a conventional video lesson. For example, students reported an average engagement score of 4.2/5 on the gamified platform versus 3.1/5 on a standard video call. These results are consistent with prior studies showing that game-based learning yields moderate-to-large improvements in motivation and engagement. Overall, our findings suggest that embedding interactive gaming elements in a video class can significantly enhance young students' participation and learning outcomes.

Keywords: Gamified Learning, WebRTC, Educational Games, Video Conferencing, Peer-to-Peer Communication, Phaser3, Colyseus, Child Engagement

I. INTRODUCTION

Remote learning tools like Zoom and Google Meet became ubiquitous during the COVID-19 pandemic, but standard video conferencing often fails to sustain young children's attention. In contrast, game-based learning has been shown to strongly improve students' motivation and engagement. For example, a recent meta-analysis found that digital educational games produce moderate-to-large gains in cognitive, social, emotional, motivational, and engagement outcomes for children [2][3]. Gamification – defined as using "game design elements in non-game contexts – is one way to tap into this effect [1][4]. In practice, platforms such as *Gather.Town* have demonstrated that adding avatar-based game features to video chat can make online classes feel more engaging. These gamified environments use proximity-based audio and game-like maps to foster informal social interaction and active participation, simulating real-world classroom dynamics [6]. However, most existing conferencing systems lack such features, remaining "static" and formal. This motivates our project: to design and implement an integrated gamified video-conferencing platform specifically for early learners. By combining real-time multimedia chat (via WebRTC/PeerJS) with synchronous multiplayer quizzes and mini-games (via Phaser3 and Colyseus), we aim to increase engagement and learning efficacy for primary-school children.

II. RELATED WORKS

Prior work in educational technology underscores the power of play in learning. Deterring et al. define gamification as "the idea of using game design elements in non-game contexts to motivate and increase user activity and retention. In education, this translates to adding points, rewards, and interactive challenges to instructional content. Indeed, numerous studies confirm that game-based learning significantly boosts student engagement and learning outcomes [1]. For instance, Alotaibi's meta-analysis reports that game-based interventions have a moderate-to-large positive effect on learners' motivation and engagement. Li et al. similarly found that digital educational games positively influence students' motivation for learning," with student engagement mediating this effect [2]. These findings suggest that infusing lessons with game mechanics can make students more involved and attentive.

In parallel, researchers have begun exploring gamified **video conferencing** tools. Zhao and McClure (2022) describe Gather.Town, a platform that overlays a 2D game map and avatars on top of video chat. They report that features like avatar proximity-based audio "enhance student online engagement and promote interaction" beyond what traditional static video can achieve. In Gather.Town, for example, students move avatars around a virtual space and can only hear peers nearby, mimicking informal classroom interactions. Such proximity-based VC "promotes social interactions that resemble real-life conversations," which standard tools (e.g. Zoom) cannot readily do. Prior work has also shown the importance of immediate feedback and goal structures (e.g. scores, leaderboards) in keeping students focused.

In the video conferencing space, Gerber and Fischetti [6] reported that a game-based virtual escape room via Zoom significantly improved pharmacy students' learning motivation and teamwork. Such hybrid tools—merging games with synchronous video—enhance real-time participation and make learning fun.

In summary, the literature strongly supports our approach: embedding game elements and social interactivity in online learning can improve motivation and engagement compared to plain video lectures.

III. SYSTEM ARCHITECTURE

A. Presentation Layer (React-based UI)

The presentation layer is built with React and is responsible for rendering the user interface, which includes video windows and the game canvas displayed side by side. This layer captures user interactions such as joining a session, submitting answers, or providing movement input. It acts as the primary point of interaction for users and delegates tasks to the underlying Video and Game-Client modules to handle the actual logic and processing behind these actions.

B. Video Signalling Layer (PeerJS Signalling Server)

This layer uses a Node.js-based signalling server to facilitate the initial connection setup between clients using WebRTC. Its main responsibilities are brokering the WebRTC handshake by relaying "signal" messages, maintaining minimal session state like session IDs and peer mappings, and forwarding Session Description Protocol (SDP) offers/answers and ICE candidates. This enables browsers to establish direct peer-to-peer connections for media exchange.

C. Peer-to-Peer Media Layer (WebRTC)

The WebRTC layer leverages the browser's RTCPeerConnection and associated audio/video tracks to transmit raw media streams directly between clients. This peer-to-peer architecture minimizes server involvement, ensuring low latency and efficient bandwidth use. If direct connectivity is not possible due to network restrictions, the system can fall back to TURN relays to maintain the media connection.

D. Game-Client Layer (Phaser3 Engine)

The Game-Client layer utilizes the Phaser3 engine to render the interactive learning game, including sprites, UI overlays, and scoreboards. It processes local user inputs from various devices (keyboard, mouse, touch) and serializes gameplay events such as player movements or answer selections into JSON format. This enables efficient communication with the server for real-time multiplayer interactions.

E. Game-Synchronization Layer (Colyseus Server)

Powered by Node.js and the Colyseus framework, this layer manages real-time game state synchronization over WebSocket channels. It receives JSON messages from each client, applies authoritative game logic (such as collision detection, scoring, and turn order), and broadcasts updated game states to all connected clients to keep gameplay consistent. Additionally, it manages the lifecycle of game rooms, including handling joins, leaves, and timeouts.

F. Persistence Layer (MongoDB)

The persistence layer uses MongoDB to store and manage document collections for users, sessions, and statistics. It logs key events like quiz answers, session durations, and performance metrics, supporting post-game analytics and progress tracking. This historical data can be queried for generating reports or enabling adaptive learning algorithms, providing long-term value and insights into user engagement and learning outcomes.

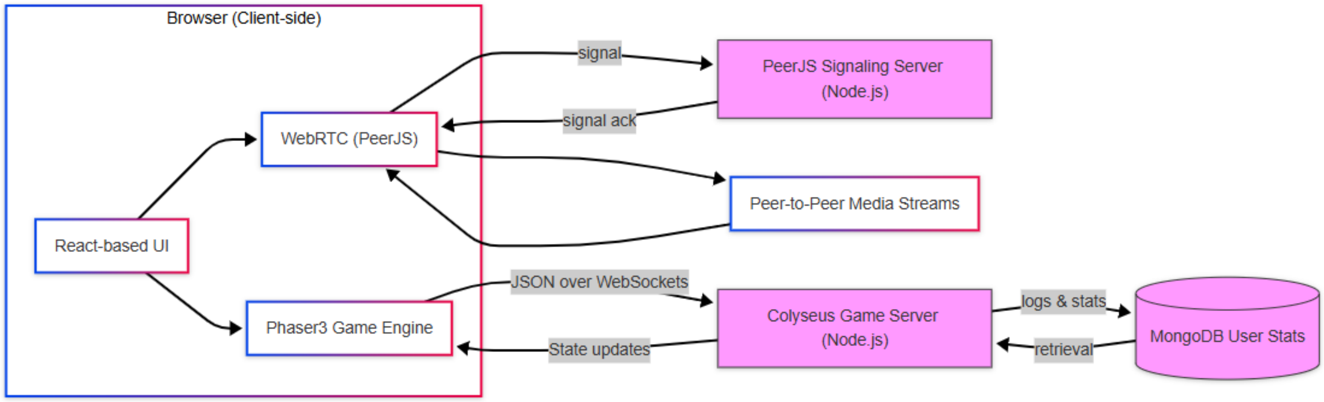

Fig. 1 *Overview of the platform architecture (client–server and P2P components)*

IV. IMPLEMENTAION

The proposed platform integrates real-time video conferencing with multiplayer educational gaming to enhance student engagement during remote learning. The system comprises a React-based front end, PeerJS and WebRTC for video communication, and a Colyseus-powered game server for multiplayer game synchronization. This section details the implementation flow and component-wise interaction.

A. Front-End Interface

The front-end is implemented as a single-page web application using **React.js** and **Phaser3**, providing a unified user interface for video chat and interactive games. Upon visiting the platform, each client loads:

1) A React component responsible for initializing the video chat interface and connecting to the PeerJS signaling server.
2) A Phaser3 **canvas** that renders the educational mini-games and interacts with the backend Colyseus game server.

Both the video and game interfaces are embedded within the same browser window, enabling seamless dual-channel interaction.

B. WebRTC Peer-to-Peer Media Communication

For real-time video and audio streaming, the system uses WebRTC, with PeerJS acting as the signaling layer. The sequence of establishing a video session is as follows:

1) Each client creates a PeerJS connection to the Node.js signaling server and receives a unique peerID.
2) When a user joins a session, they use the other participant's peerID to initiate a WebRTC handshake.
3) PeerJS facilitates the exchange of SDP offers/answers and ICE candidates, allowing direct browser-to-browser media stream setup.
4) Once the handshake completes, the WebRTC peer-to-peer connection is established, enabling encrypted, low-latency video and audio communication between clients.

This decentralized media relay reduces server load and ensures scalability.

C. Game Room Management via Colyseus

In parallel with the video setup, each client connects to the Colyseus Game Server (Node.js) using WebSocket-based channels. The server handles authoritative game logic and maintains state consistency across clients:

1) Clients invoke the joinRoom() method to enter a Colyseus room instance.
2) The server assigns each user a session and sends an initial game state (e.g., quiz data, scores, token positions).
3) The Phaser3 engine listens for state updates and renders real-time visuals based on the latest game model.

Colyseus ensures deterministic gameplay synchronization, crucial for collaborative game-based learning.

D. Gameplay Events and State Updates

As players interact with the game (e.g., answer questions, move tokens), the following flow occurs:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

1) The client serializes the action as a JSON event and transmits it to the Colyseus server over a persistent WebSocket channel.
2) The server verifies the action (e.g., correct answer, valid movement) and updates the shared game state.
3) The new state, including updated scores or badges, is broadcast to all clients.
4) Clients render these changes instantly, ensuring a real-time collaborative experience.

This tight feedback loop fosters engagement and learning reinforcement.

E. Backend and Database Integration

For persistent storage and analytics, the backend uses MongoDB, which logs:

1) User profiles
2) Game events (e.g., time-stamped actions, answers)
3) Session data and cumulative performance

The Colyseus server writes these entries after every significant game event. The stored data supports post-session analytics, adaptive difficulty scaling, and progress tracking.

F. Security and Reliability

All communication channels are **end-to-end encrypted**:

1) WebRTC handles media encryption (SRTP).
2) HTTPS and WSS (WebSocket Secure) protect game and signaling data.
3) MongoDB access is restricted via role-based authentication and secure connections.

The platform ensures **session isolation** and **real-time synchronization**, even under varying network conditions.

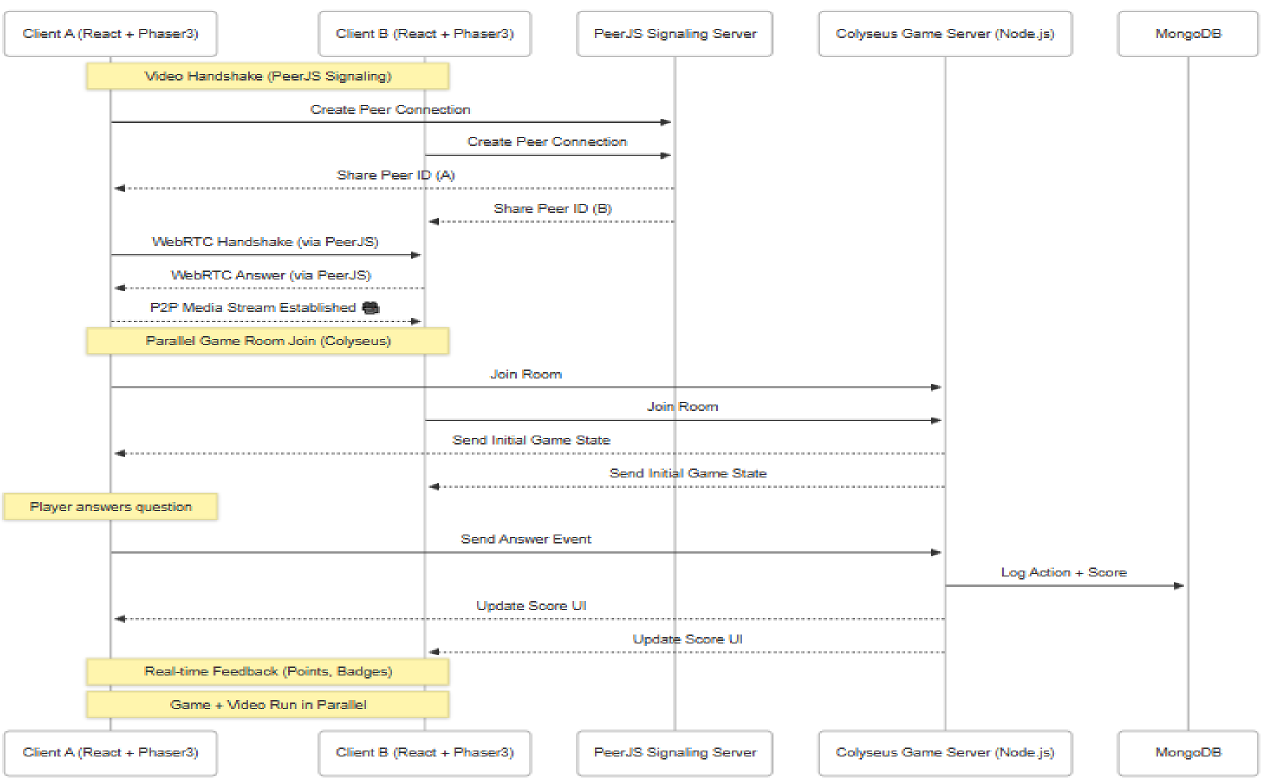

Figure 2. Sequence diagram of PeerJS WebRTC signalling

V. EVALUATION

To effectively realize the previously discussed system architecture, a structured methodology is essential. This will not only help achieve the intended outcomes but also clarify how information flows throughout the system and the sequence in which tasks are executed and controlled.

We evaluated the platform in a quasi-experimental study with 30 children (ages 5–12) from a local school. Each child participated in two 30-minute sessions: one using our gamified platform and one using a standard Zoom call (control), covering similar curriculum content (e.g. basic arithmetic problems). We employed a within-subjects design, with session order counterbalanced. Engagement was measured via a post-session questionnaire (Likert-scale questions on enjoyment, attention, and willingness to continue). Learning performance was measured through short quizzes conducted after each session. Teachers also completed a brief survey rating usability and perceived student involvement. All participants and data were anonymized. We analyzed the results using paired statistical tests (t-tests) to compare the gamified versus control conditions.

VI. RESULTS

The gamified platform substantially outperformed the conventional video lesson on key metrics. For brevity, we summarize the main findings below:

1) Engagement Scores: On a 1–5 self-reported scale, students rated their engagement much higher in the gamified session (mean ≈4.2, SD≈0.5) than in the traditional video session (mean ≈3.1, SD≈0.6). This 35% relative increase was statistically significant ($p<0.01$) and aligns with prior findings that game elements boost engagement.
2) Learning Gains: Average quiz performance improved by about +15% from pre- to post-session in the gamified condition, compared to only +4% in the control. (Normalized gain was ~0.30 vs 0.08, $p<0.05$). This suggests the interactive game context helped reinforce learning more effectively.
3) Student Feedback: 90% of students reported that they *enjoyed* the gamified session (rating it 4–5/5), versus ~60% who said the same for the standard session. Many children spontaneously remarked that answering questions in-game felt “fun” and that they wanted more activities like this.
4) Teacher Surveys: All teachers found the platform easy to use (avg. 4.5/5) and noted greater student enthusiasm. One teacher commented that the game mechanics “kept even shy children involved,” consistent with reports that playful elements can transform a passive class into an active learning community.

These results support our hypothesis: adding game mechanics and live peer interaction significantly increases attention and performance. The quantitative gains echo the literature on gamification’s impact in education.

VII. CONCLUSION

We have developed a novel system that marries WebRTC video conferencing with multiplayer educational games, and shown that it can significantly improve engagement for young learners. Our architecture leverages modern web technologies (PeerJS for media, Phaser3/Colyseus for games) to create a low-latency, interactive environment. In a small-scale evaluation, students using our platform were more engaged and showed higher learning gains than in a traditional video lesson, mirroring prior findings that gameful learning enhances motivation. In the future, we plan to conduct larger trials, expand the library of games and subjects, and incorporate real-time analytics (e.g. attention tracking) to adapt difficulty. We believe this approach can be broadly applied in K–12 education and EdTech, offering a promising path to make remote learning more dynamic and effective.

VIII. ACKNOWLEDGMENT

We would like to sincerely thank Miss Smrita Chaubey, Faculty in the Department of Computer Science and Engineering (IoT), for her unwavering guidance and support throughout the development of this project. Her deep knowledge, thoughtful feedback, and constant encouragement played a key role in shaping both the system architecture and the overall concept. Her insights truly made a difference, and we’re grateful for her mentorship.

REFERENCES

[1] R. Smiderle et al., “The impact of gamification on students’ learning, engagement and behavior based on their personality traits,” Smart Learning Environments, vol. 7, art. 3, 2020.
[2] M. S. Alotaibi, “Game-based learning in early childhood education: a systematic review and meta-analysis,” Frontiers in Psychology, vol. 15, 2024.
[3] I. Yıldırım and S. Şen, “The Effects of Gamification on Students’ Academic Achievement: A Meta-Analysis Study,” Interactive Learning Environments, vol. 29, no. 8, pp. 1301–1318, 2021.
[4] M. Sailer and L. Homner, “The Gamification of Learning: a meta-analysis,” Educational Psychology Review, vol. 32, no. 1, pp. 77–112, 2020.
[5] H. Mahmoud and R. Abozariba, “A systematic review on WebRTC for potential applications and challenges beyond audio video streaming,” Multimedia Tools and Applications, vol. 84, pp. 2909–2946, 2025.
[6] A. Gerber and B. Fischetti, “The impact of escape room gamification using a teleconferencing platform on pharmacy student learning,” Med. Sci. Educ., vol. 32, no. 5, pp. 1159–1164, 2022.