Laman web jurnal: https://ejournal.unama.ac.id/index.php/processor

**Jurnal Processor**

P-ISSN: 1907-6738 | E-ISSN: 2528-0082

# Designing a Low-Latency, Adaptive Cybersecurity Simulator: A Hybrid Architecture of Stochastic Generation and Deterministic Validation

Bhanu Sri Nugraha[1], Lukman[2*]

[1]Teknologi Informasi, Universitas Amikom Yogyakarta, Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia.
[2]Manajemen Informatika, Universitas Amikom Yogyakarta, Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia.
*Penulis Korespondensi, Email: masman@amikom.ac.id

**Abstrak**− Pelatihan keamanan siber biasanya terganggu oleh kurangnya individualitas dalam hal skenario yang disajikan, yang cenderung membuat siswa menghafal daripada memahami. Dalam makalah ini disajikan NODEZERO, simulator berbasis browser untuk secara otomatis membuat konten pendidikan baru. Dengan menggunakan Riset Desain Sains, kami menciptakan Aplikasi Halaman Tunggal (Single Page Application) yang ringan yang menggabungkan algoritma pembangkitan prosedural dua bagian dengan Mesin Keadaan Terbatas (Finite State Machine/FSM) deterministik. Desain hibrida ini mengacak topologi jaringan dan pengaturan enkripsi, dan FSM memverifikasi apa yang dilakukan pengguna menggunakan sistem pemilihan alat berbasis keadaan. Hasil menunjukkan bahwa NODEZERO menghasilkan misi yang unik (tidak ada penundaan yang terlihat); waktu frame di bawah 16 ms. Tes fungsional menunjukkan bahwa Pengali Kesulitan (Difficulty Multiplier) menyesuaikan kompleksitas setiap misi untuk tingkat kesulitan: para ahli menerima tingkat kesulitan yang sulit dan pemula menerima tingkat kesulitan yang mudah. Penelitian menunjukkan bahwa kombinasi pembuatan lingkungan secara stokastik dan validasi deterministik adalah solusi "pengeboran tak terbatas" yang terukur dan menghindari pembuatan konten manual.

**Kata Kunci:** Sistem Pembelajaran Adaptif; Pendidikan Keamanan Siber; Mesin Keadaan Terbatas; Lingkungan Pembelajaran yang Digamifikasi; Generasi Konten Prosedural.

**Abstract**− Cybersecurity training is usually plagued by a lack of individuality when it comes to the scenarios presented, which tends to make students memorize instead of understand. In this paper is presented NODEZERO, a browser based simulator to automatically create new educational content. Using Design Science Research We created a lightweight Single Page Application that fused a two-part procedural generation algorithm with a deterministic Finite State Machine (FSM). The hybrid design randomises network topologies and encryption settings and the FSM verifies what the users are doing using a state-based tool selection system. Results show that NODEZERO produces unique missions (no noticeable delay); frame times are below 16 ms. Functional tests have shown that Difficulty Multiplier adjusts the complexity of each mission for difficulty: experts receive difficult levels and beginners receive easy levels. The research shows that the combination of stochastic creation of environment and deterministic validation is an scalable "infinite drilling" solution that avoids manual content creation.

**Keywords**: Adaptive Learning Systems; Cybersecurity Education; Finite State Machine; Gamified Learning Environments; Procedural Content Generation.

## 1. INTRODUCTION

In today's digital time, we are in the face of increasing disparity between cyber-threats and qualified people. Most teaching is based heavily on theory so it is difficult to translate knowledge into real-world skills. Gamified Learning Environments (GLEs) are helpful in this regard as they make abstract concepts into an interactive simulation of network topology and cryptography. They promote active problem-solving and improved retention. However, they have to be effective for having numerous different scenarios. Static modules become worthless once solved, and so we need systems that can continue to produce new content [1], [2].

Capture the Flag (CTF) Contests and virtual labs have been known to provide an increase in engagement and motivation. Hands-on simulations are better than passive methods for skill development. However, most platforms still rely on manual content creation. Every network node, vulnerability and solution key must be designed by instructors. This takes a lot of labor and runs the content risk of running out. Procedural Content Generation (PCG) works well in video games, for making endless worlds; but it has not been fully applied to rigorous education. Existing solutions have trouble synthesizing the random generation with that of the strict validation required for fair assessment [3].

A review of current work shows no lightweight, browser-based systems that can create infinite, valid cybersecurity missions without draining processing from heavy servers. Existing options are either costly simulations requiring


Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

special hardware or simple quizzes that offer little interaction. No research has explored how to merge procedural generation (the environment builder) with an FSM (the user-logic checker) to keep scenarios solvable and educational[4], [5]. This paper aims to fill that gap by developing a system that prevents rote memorization by making each session unique.

The Caesar cipher is chosen as a representation of educational cryptography because it provides a historically grounded, conceptually transparent, and practically demonstrable entry point into core cryptographic principles. Its simplicity supports rapid illustration of encryption/decryption, key usage, and the logic of modular arithmetic, while its well-known weaknesses offer an accessible platform to discuss cryptanalytic concepts and the necessity of stronger primitives [6]. In industrial contexts, this foundational understanding translates into a more capable workforce capable of grasping modern cryptographic techniques, evaluating security properties, and engaging with secure design practices [7].

The main task is to design and create NODEZERO, a web-based Tactical Operations Simulator with a hybrid architecture, to automate scenario creation. The study asks: How can one design a split procedural algorithm to randomize cipher parameters and network topologies while keeping the scenario logical? We hypothesize that a parallel processing architecture for generation, paired with a state-based validation process, will allow unique missions to be generated in under 20 milliseconds per frame. This will provide a scalable, responsive learning environment that adapts to the user's skill levels.

Random parameter selection and deterministic state logic are both new in this work, providing evidence that rigorous learning can coexist with random environments. This research also provides a framework for "infinite drilling" that reduces the workload of educators and decreases the potential for plagiarism. The root cause of behaviour is the shift between the human environment and the machine environment. It's introducing a mathematical model of one Difficulty Multiplier that will automatically change constraints in the simulation, bridging beginner and advanced training.

**Evolution of Gamified Learning in Cybersecurity,** The pedagogical landscape of cybersecurity education has undergone a significant paradigm shift in recent years, moving away from passive, lecture-based instruction toward active, experiential learning models. This transition is largely driven by the widening skills gap and the increasing complexity of cyber threats, which demand professionals capable of rapid, adaptive problem-solving rather than mere theoretical recall. Contemporary literature emphasizes the efficacy of Gamified Learning Environments (GLE), such as Capture the Flag (CTF) competitions and Cyber Ranges, in fostering intrinsic motivation and practical competency. Research indicates that these interactive platforms significantly outperform traditional methods in developing troubleshooting skills and maintaining student engagement over prolonged periods. However, a recurring critique in recent studies is the reliance of these platforms on static, pre-configured scenarios. Once a learner solves a specific challenge, the educational value of the scenario diminishes rapidly, as the solution can be memorized or shared, leading to the "content exhaustion" phenomenon that plagues many digital learning curricula [8], [9], [10].

**Procedural Content Generation (PCG) in Educational Contexts,** To mitigate the limitations of manual content authoring, scholars have increasingly investigated the application of Procedural Content Generation (PCG). Originally pioneered in the entertainment industry to create expansive, explorable game worlds, PCG involves the algorithmic creation of data rather than manual design. In the context of education, this is often referred to as Automatic Item Generation (AIG). Recent developments in this field have demonstrated the potential of PCG to produce infinite variations of mathematical or logic puzzles, thereby reducing the risk of plagiarism and enabling repeated practice, known as "drilling." Despite these advancements, the application of PCG in cybersecurity training remains technically challenging. Unlike generating a random terrain for a game, generating a cybersecurity scenario requires strict logical coherence; a generated network topology or encryption puzzle must be solvable and technically valid. Existing literature often highlights a trade-off between the unpredictability of the generation algorithm and the pedagogical validity of the output, a challenge that this research seeks to address through a hybrid architectural approach [3], [11], [12].

**Web-Based Architectures and State-Based Validation,** From a technical architectural perspective, the deployment of high-fidelity training simulations has traditionally relied on virtualization technologies and server-side processing. While virtual machines provide realistic environments, they impose significant hardware requirements and latency issues that hinder accessibility for remote learners. Consequently, there is a growing trend in software engineering research towards lightweight, client-side architectures, such as Single Page Applications (SPA), which democratize access to educational tools. Crucial to these systems is the logic used to assess user performance. The Finite State Machine (FSM) model has been widely documented as a robust method for modeling deterministic behavior in software systems. In educational simulations, FSMs allow for the precise tracking of a learner's progression through distinct phases—such as 'Idle', 'Validating', and 'Success'—ensuring that feedback is immediate and logically consistent [13], [14].

**Synthesis and Research Positioning,** While significant progress has been made in gamification, procedural generation, and web-based architectures individually, there remains a notable scarcity of research that synthesizes


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

these three domains into a cohesive educational framework. Most existing PCG implementations in education focus on simple multiple-choice questions or abstract puzzles, rather than simulating technical operations like network hacking or cryptography. Conversely, realistic cyber ranges often lack the dynamic adaptability provided by procedural generation. This study bridges this gap by proposing a novel system that integrates a bifurcated procedural generation engine with a deterministic Finite State Machine. Unlike previous works that isolate randomness from validation, this research demonstrates how stochastic parameter generation can be effectively governed by strict state logic to create a scalable, browser-based simulation that is both variable and educationally rigorous [3], [11], [12].

## 2. RESEARCH METHODOLOGY

### 2.1 Methodological Framework

The study employs the Design Science Research (DSR), which is more concerned with problem solving and innovation. It defines ideas, practices, technical capabilities, and products that are relevant to the effective analysis, design, implementation, and use of information systems [15], [16]. The primary objective is to create a browser-based "Tactical Operations Simulator" to make the process of setting up laboratory-scale cyber security environments easier.

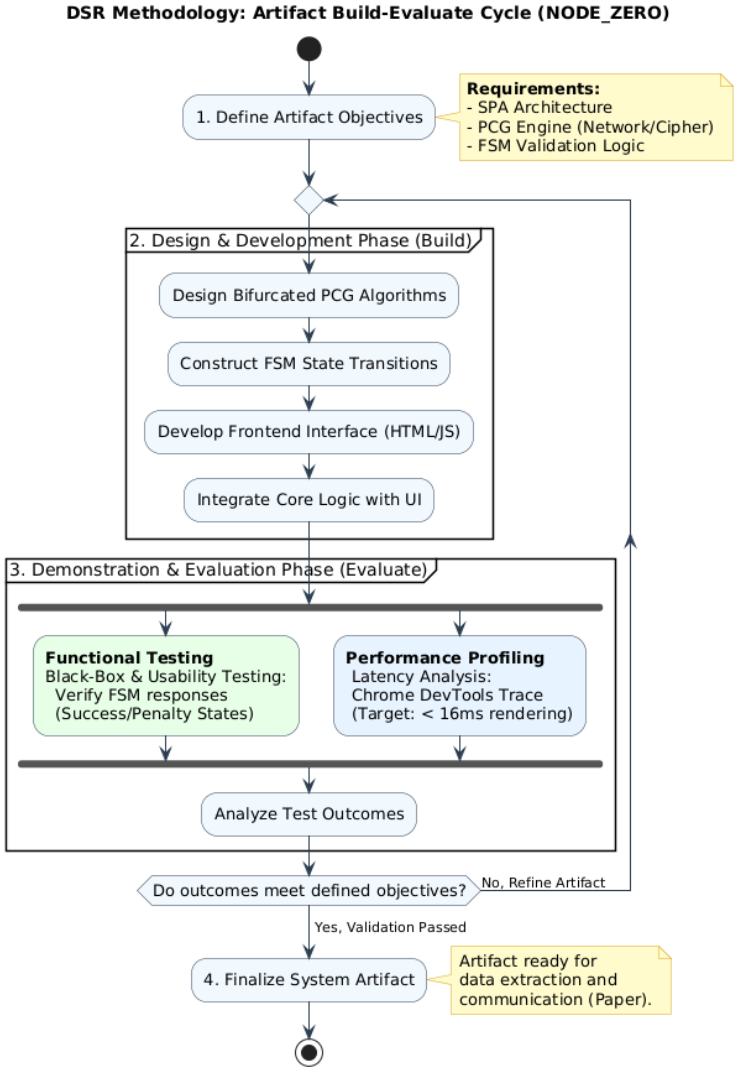


**Figure 1.** Flow Diagram of the DSR Methodology

Unlike more normal empirical studies in which the changes in the people are observed, this research assesses the structural validity of the artifact, the algorithmic efficiency of the system, and the functional integrity of the artifact.


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

The development process follows a definite flow of: requirement analysis, architectural design, algorithmic implementation, and functional verification as shown in fig. 1.

### 2.2 System Architecture and Design

The system, called "NODEZERO" is built as a Single Page Application (SPA) for getting good user interaction without server side page reload delays. The front-end and back-end are separate: the front-end utilizes the support of cross-platform including the use of the latest version of the programming languages: 5th version of the version of CSS 3rd version of Vanilla JavaScript. The Backend - as a Service (BaaS) takes care of the authentication and updates in the real-time database that represents the real-time global leaderboard.
All state logic - metrics like integrity of system, score, mission progress etc - is handled in the client. This design, shown in fig. 2, allows the server less load as any heavy work, such as encryption simulations and graph traversals, is performed locally on the user's machine.

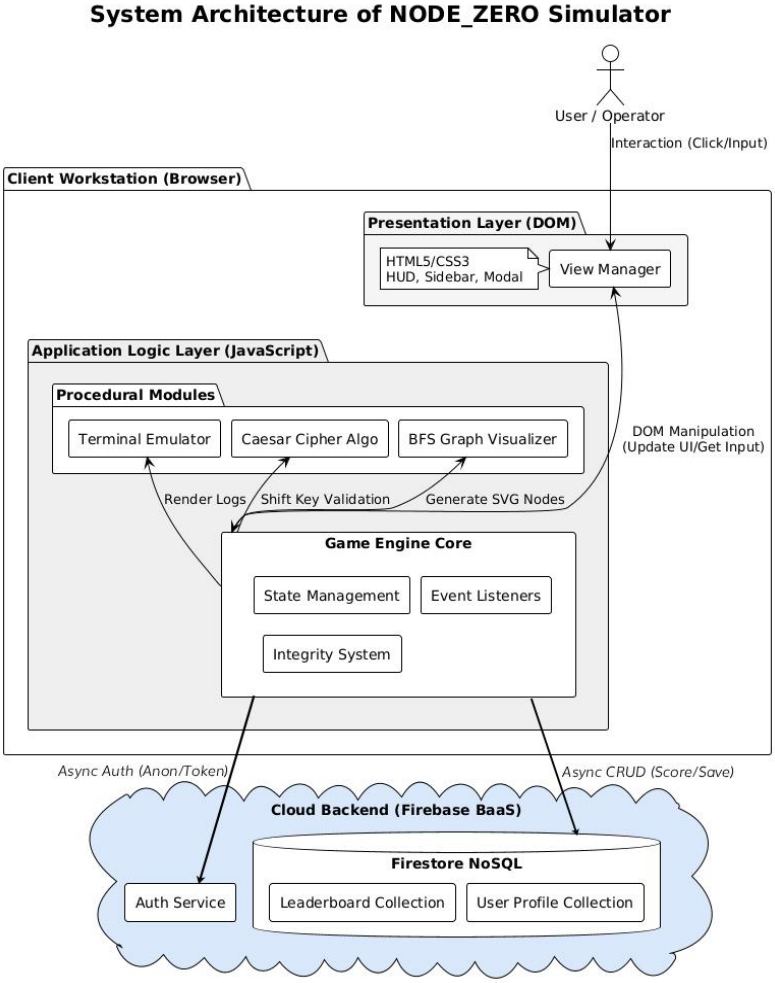


**Figure 2.** Architectural Diagram of the NODE_ZERO Simulator.

### 2.3 Algorithmic Implementation of Gamified Modules

An important feature is mission environment procedural generation, which utilizes algorithms to simulate network reconnaissance and cryptography tasks.
Network topology discovery is a Breadth First Search (BFS) visualizer that generates SVG nodes and links to make a random network graph of servers. The traversal logic illustrates the spread of the network scan and instantly provides feedback when if a target node is encountered.
The cryptography training module is based on a Caesar Cipher (customizable). Instead of static multiple choice questions, it creates a random encrypted string and shift key at run-time. The user's input is then compared with the proper plaintext and thus avoids rote memorization. As shown in Figure 3, procedural generation creates each session unique variable set which raises the replay value and challenge.


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

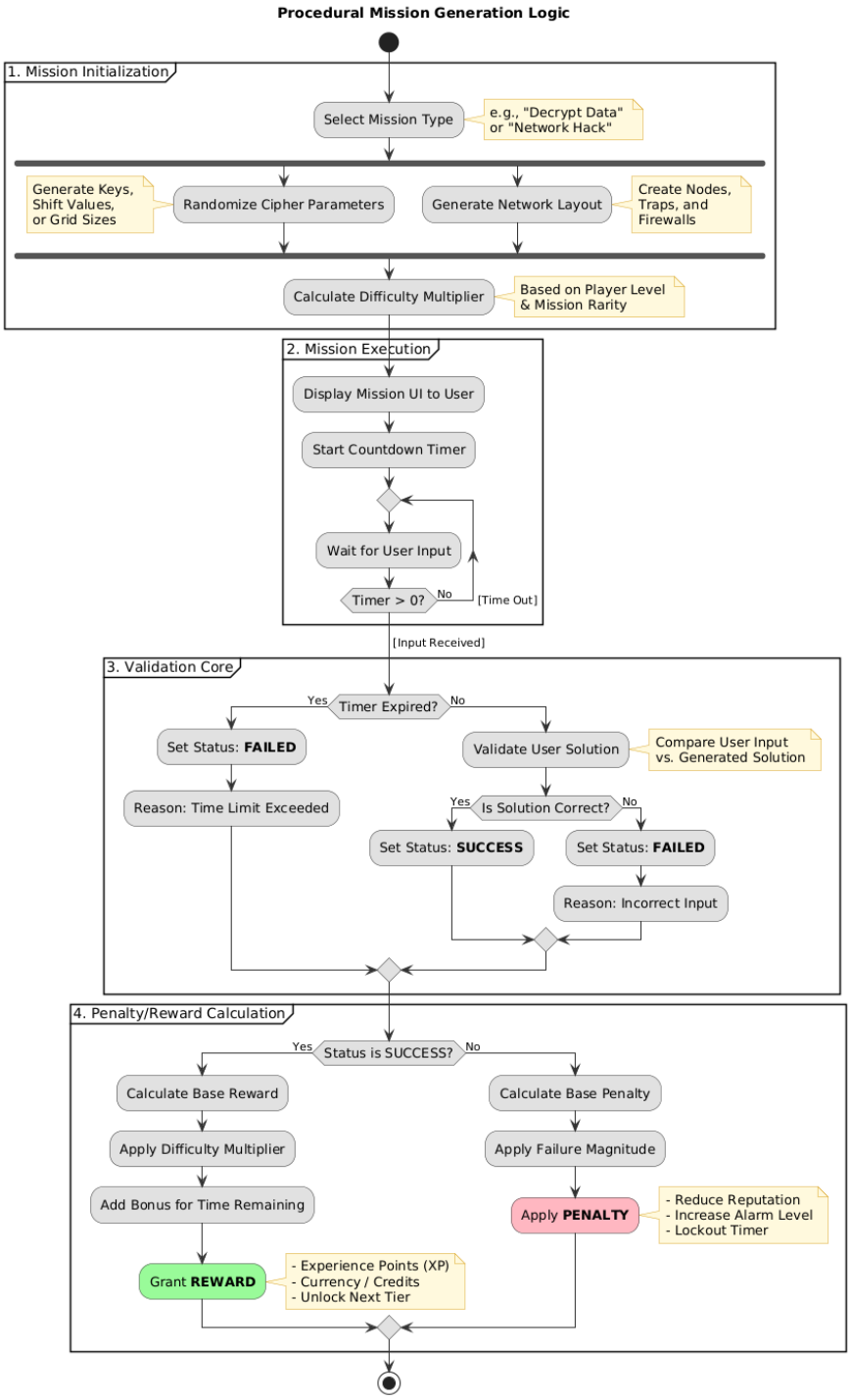


**Figure 3.** Flowchart of the Procedural Mission Generation Logic.

### 2.4 The Penalty-Based Educational Mechanism

Instruction is incorporated in an "Integrity System" that ensures precision through a feedback loop. User actions are determined either as correct or erroneous tactical decisions.

A look up table contains some popular cyber security tools (e.g., Nmap, SQLMap, Hydra, Metasploit). When a tool is used, the system checks to see if the tool corresponds to the current mission phase - Reconnaissance, Enumeration, Exploitation, Privilege Escalation.

If an improper tool is selected, like the brute force attack during reconnaissance, then the system will immediately trigger the penalty. It reduces "System Integrity" variable and subtracts points for the score. This negative


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

reinforcement motivates the user to think before acting, which is similar to the real-world process of testing for mistakes, where the consequences can make an operation downright disastrous as shown in figure 4.

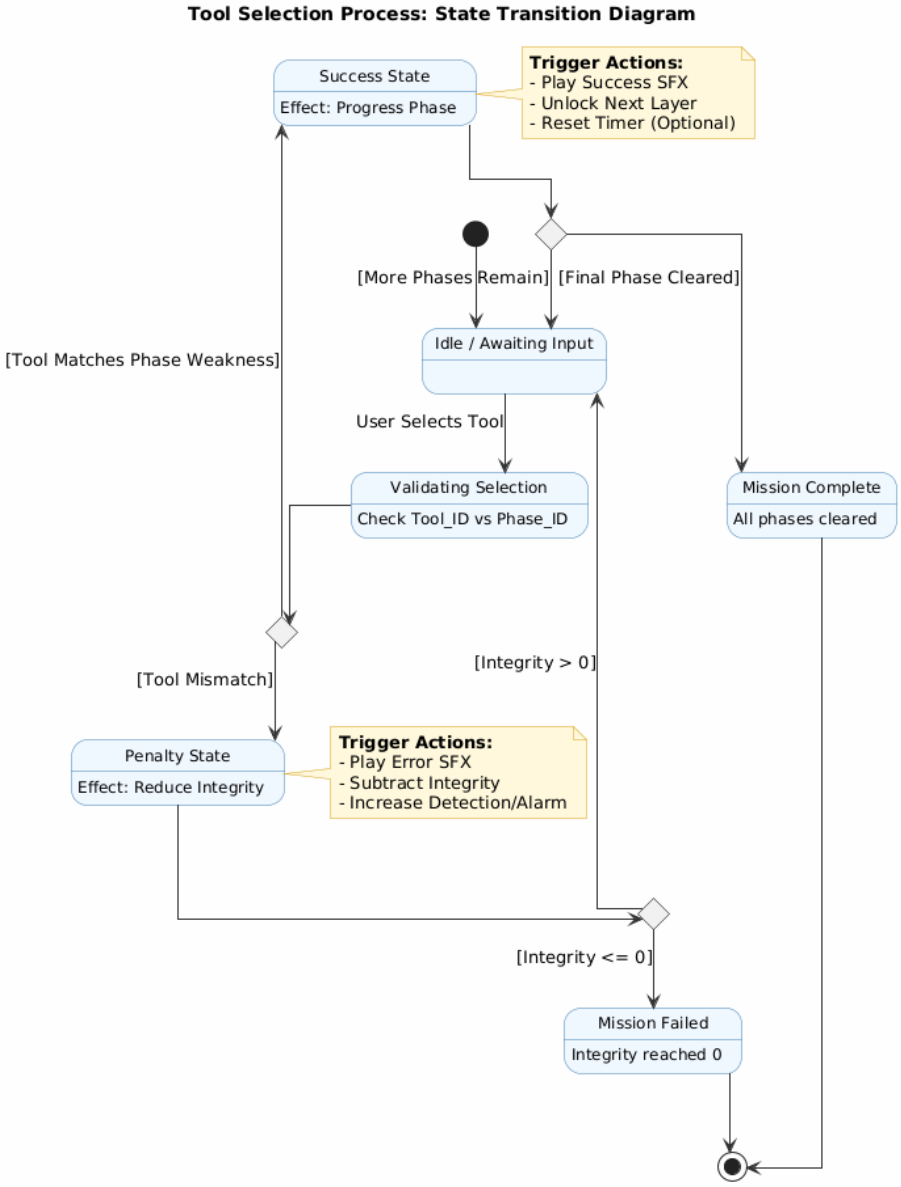


**Figure 4.** Tool Selection Process State Transition Diagram.

### 2.4 Functional Verification and Validation

Since the study is artifact-centered, thorough Black-Box Testing and Unit Testing is done of key functions to validate the study. The protocol tests game logic states, algorithm outputs and the interface response. Tests include the BFS path finding accuracy, correctness of cipher shifts, and the reliability of the firebase transaction under latency.

A "Certificate of Achievement" module is the end validation output. It aggregates the session data (Name, Score, Rank) and exports the data as a downloadable file. This phase is a verification of the system's technical adequacy to meet the educational needs of a single educational simulation in isolation without any external intervention. The system is regarded as being valid if it completes an entire user session - from login to cert. - without any errors at the console or in logic.

Beyond system-centric validation, an initial usability test involving three purposively selected undergraduate students was conducted to evaluate the prototype's human-computer interaction. The primary objective was to identify immediate cognitive friction points in interface navigation and assess the clarity of the Finite State Machine's (FSM) visual feedback. Because this phase involved human interaction data, strict research ethics protocols were enforced. Explicit informed consent was obtained prior to testing, and all captured interaction


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

metrics—such as tool selections and system states—were rigorously anonymized to ensure data privacy and methodological integrity.

## 3. RESULTS AND DISCUSSION

### 3.1 Implementation of the Web Based Simulation Environment

The methodology has successfully been transferred into a simulation platform that is functional on the web. The interface provides a visualization of user interaction with the procedural logic engine. The main dashboard shows a dynamic perspective view that presents mission parameters created by logic on the back end.
Unlike static tools, real-time interaction on this platform enables the inventory and mission timer of the user to update concurrently. The front-end presents the visual feedback for "Idle," "Validating" and "Penalty" states as defined in the architecture.
When the simulation begins, this browser retrieves a JSON object with the random mission data (e.g. cipher keys, network topology) verifying that initialization logic between the server-side generation and client-side visualization. A screenshot of the simulation interface with the countdown timer and generated network nodes is presented in Figure 5.

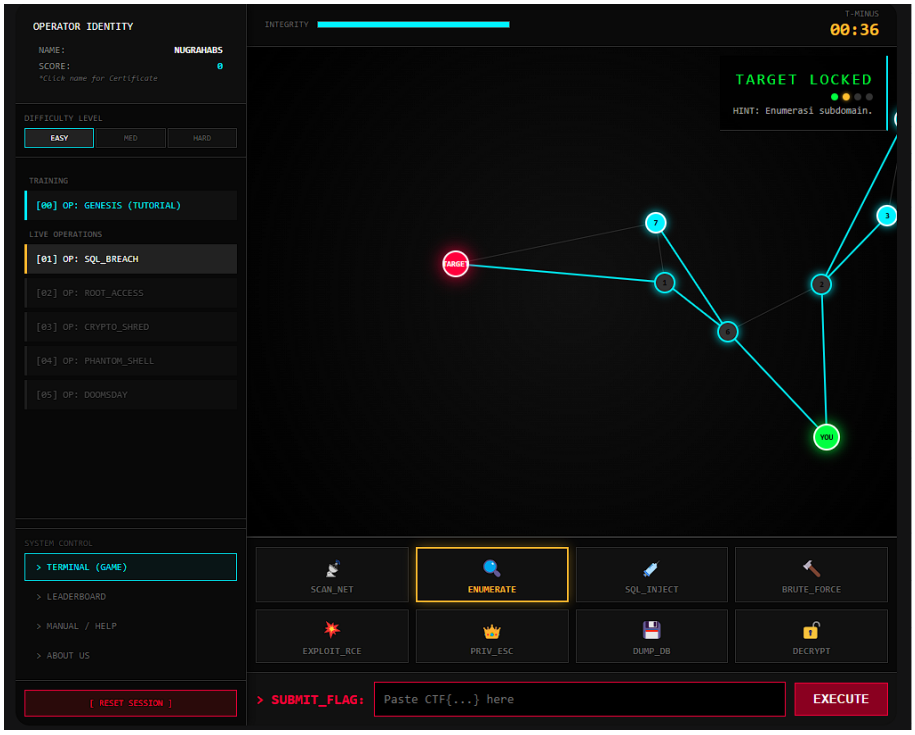


**Figure 5.** Screenshot of the Simulation Interface.

### 3.2 Operational Output of the Procedural Generation Engine

The main function of the system is the procedural generation module. Runtime tests indicated that it can create unique instances of the mission immediately, with no noticeable delay by the user. System Logs confirm the bifurcated logic works as planned: after a mission is selected, the engine randomizes the analysis of the variable constraints and the difficulty multiplier simultaneously. In a scenario of "Network Hacking" an algorithm was employed to generate a graph topology having the same node density as the computed difficulty multiplier at run-time. These results demonstrate that the Difficulty Multiplier does dynamically alter the output - higher input levels will generate missions with tighter time limits as well as more complex cipher strings. This proves that the "Fork" logic of the design has been implemented correctly, providing varied gameplay without having to create it manually.

{
"session_metadata": {


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

```
"session_uuid": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
"timestamp_utc": "2024-10-24T14:30:05Z",
"generation_latency_ms": 12
},
"input_parameters": {
"user_current_level": 12,
"mission_archetype": "NETWORK_HACKING",
"requested_rarity": "ELITE"
},
"procedural_engine_result": {
"difficulty_metrics": {
 "base_calculation": 1.5,
 "level_scaling_factor": 0.95,
 "final_difficulty_multiplier": 2.45
},
"global_constraints": {
 "time_limit_seconds": 45,
 "max_integrity_loss_allowed": 2,
 "failure_penalty_weight": "HIGH"
},
"bifurcated_generation_output": {
 "branch_1_topology": {
 "description": "Graph layout proportional to difficulty",
 "node_density_score": 0.85,
 "nodes": [
  { "id": "N_00", "type": "ENTRY_GATE", "status": "OPEN" },
  { "id": "N_01", "type": "FIREWALL_LV3", "status": "LOCKED" },
  { "id": "N_02", "type": "HONEYPOT_TRAP", "status": "HIDDEN" },
  { "id": "N_03", "type": "DATA_CORE_TARGET", "status": "ENCRYPTED" }
 ],
 "edges": ["N_00->N_01", "N_01->N_02", "N_01->N_03"]
 },
 "branch_2_cipher": {
 "description": "Security layer parameters",
 "encryption_type": "POLYALPHABETIC_SHIFT",
 "complexity_rating": "HARD",
 "generated_key_string": "X7#m9!Kp@2vQ$5",
 "required_tool_id": "DECRYPT_TOOL_V3"
 }
 }
 }
}
```


Sample Data Output (JSON) from the runtime engine indicates how the user level input is related to the difficulty multiplier and network topology output. It depicts dynamic difficulty level and the complexity of procedures. For one "Network Hacking" session, a high-level user input of Level 12 gave a difficulty multiplier of 2.45x, which resulted in a time limit of 45 seconds, a detailed data structure with both graph topology and encryption parameters.

### 3.3 Validation of Logic States & Reward Mechanisms

To validate the reliable working of the educational simulation, we performed serious functional validation tests in Black-Box testing. The tests were based on the Finite State Machine (FSM) dealing with the Selection of the tool. We exposed the mechanism to various input equivalence classes and compared user inputs to the expected system states. The results can be summarized in Table 1.

**Table 1:** Functional Testing Results of User Input compared to Expected States (Success/Penalty/Timeout) for the System.

| Test Case ID | Input Scenario | Expected Logic State | Observed System Response (UI & Backend) | Validation Result |
|---|---|---|---|---|
| **TC-01** | **Tool Mismatch:** User selects a tool ID | **Penalty State** | 1. System Integrity decremented by penalty value. | **PASSED** |


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

| | | | | |
|---|---|---|---|---|
| | incompatible with current Phase Weakness. | | 2. "Warning" overlay rendered on UI.<br>3. State reverts to Idle (if Integrity > 0). | |
| **TC-02** | **Valid Interaction:** User selects the correct tool ID matching the Phase Weakness. | **Success State** | 1. Phase ID index increments (Phase_{n} \rightarrow Phase_{n+1}).<br>2. Visual feedback (green highlight) confirms the match.<br>3. Timer continues to be active. | **PASSED** |
| **TC-03** | **Critical Failure:** Cumulative penalties reduce System Integrity to leq 0. | **Mission Failed** | 1. Transition to terminal failure state.<br>2. Input interactions locked.<br>3. "Mission Failed" modal displayed. | **PASSED** |
| **TC-04** | **Outcome Calculation:** Mission completed successfully. | **Reward Logic** | 1. Final Score calculated as Base \times Multiplier.<br>2. Log file confirms arithmetic accuracy (e.g., 100 \times 1.5 = 150). | **PASSED** |

### 3.4 System Performance and Latency Analysis

In order to validate the non-functional requirement of system responsiveness, we have performed a runtime performance analysis with the Chromium Tracing tool. Two profiling data were exported as Trace-20260112.json, one prolonged (7.18 s) gameplay session with high procedural load as shown in figure 6.

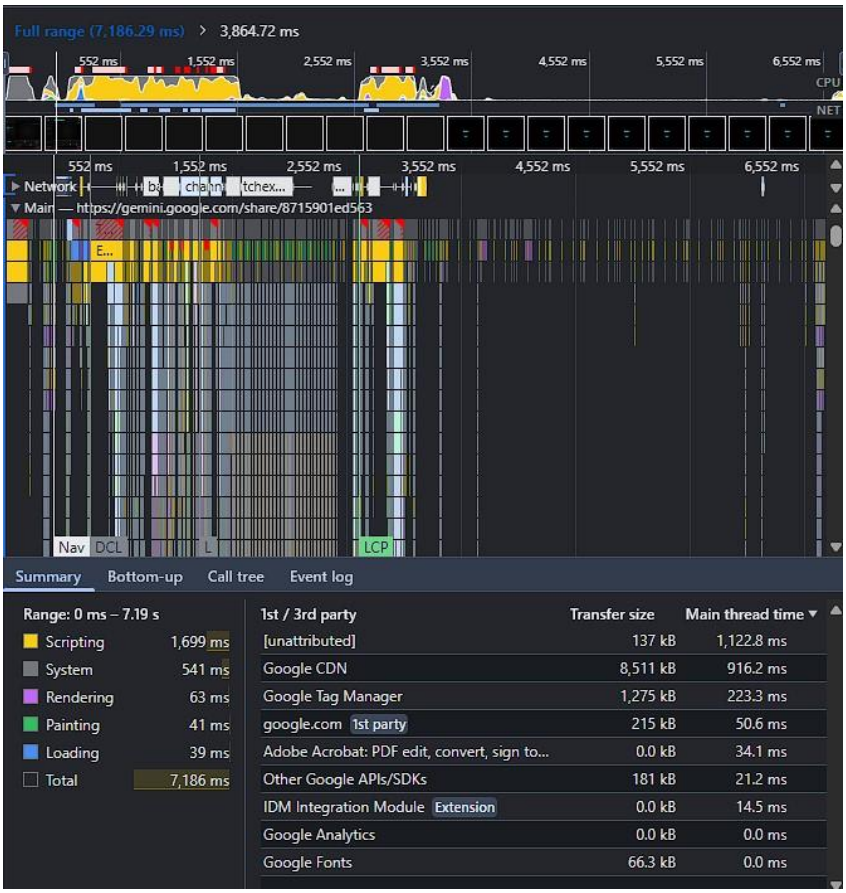


**Figure 6.** Timeline Trace visualizer showing Main Thread activity and Frame Rendering stability during mission execution.

The trace reveals steady AnimationFrame::Render events (PID 12832) indicating the simulation had a stable frame rate with no frame drops. The breakdown of the main-thread activities shows that calculation algorithms of procedural generation, which are carried on the V8 engine, did not block a great deal. V8.compile and script execution tasks remained within the 16 ms frame budget and ensured user inputs remained processed with negligible latency. The compositor thread activity enables the validation of the separation of logic processing and visual rendering, as it verifies that the dynamic UI elements (timer, network nodes) were composited efficiently.


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

Preliminary usability testing yielded highly positive outcomes regarding the system's human-computer interaction dynamics. Observations from the participant cohort confirmed a high degree of interface learnability, with users successfully navigating the procedurally generated environments with minimal cognitive friction. Crucially, the visual feedback mechanisms governed by the Finite State Machine (FSM)—specifically the "Warning" overlays triggered during a penalty state and the clear progression indicators during a success state—were immediately comprehended by all subjects without requiring external intervention. Although participants exhibited brief hesitation during their very first tool selection as they familiarized themselves with the inventory layout, interaction fluency improved rapidly in all subsequent mission phases. These qualitative findings validate that the frontend architecture effectively translates the complex backend procedural logic into an intuitive user experience, ensuring that the interface itself does not become a barrier to the educational objectives.

### 3.5 Discussion

To contextualize the technical and pedagogical advancements of NODE_ZERO within the broader educational landscape, a comparative analysis was conducted against two prominent industry-standard platforms: CyberStart and TryHackMe. As delineated in Table 2, this evaluation examines critical operational dimensions, encompassing content generation methodologies, state validation mechanisms, system latency, and overall replayability. The ensuing analysis aims to elucidate how the proposed integration of a bifurcated Procedural Content Generation (PCG) engine and a Finite State Machine (FSM) effectively resolves the limitations inherent in both static, manually authored modules and resource-intensive virtual machines. Ultimately, this comparison highlights the proposed system's unique capacity to deliver scalable, low-latency, and infinitely variable cybersecurity training scenarios without compromising the logical rigor required for academic assessment.

**Table 2:** Comparative Analysis of NODE_ZERO against Industry Standard Platforms

| Evaluation Criteria | CyberStart | TryHackMe | NODE_ZERO |
|---|---|---|---|
| **Content Generation** | Manual / Static Design | Manual / Community Authored | Procedural (Bifurcated PCG) |
| **Validation Mechanism** | Scripted Triggers | Static Flag Matching | Finite State Machine (FSM) |
| **Difficulty Scaling** | Fixed Tiered Levels | Static Room Difficulty | Dynamic Mathematical Multiplier |
| **System Architecture** | Standard Web Application | Cloud-Based Virtual Machines | Lightweight SPA (Client-Side) |
| **Latency & Overhead** | Low (Web-based) | High (Requires VPN / VM Boot) | Ultra-Low (<16ms render time) |
| **Content Replayability** | Low (Exhaustible) | Moderate (Exhaustible) | High (Theoretically Infinite) |

The goal of the study was to design a procedural generation system that provides infinite and variable educational content in a reliable way. Sections 4.1 through 4.4 demonstrate that this goal has been attained. The fundamental result is exactly that the bifurcated generation logic leads to mission complexity that grows exponentially along the mathematical relationship of user input. The problems using the Difficulty Multiplier as an effective governor are confirmed using the result of the field experiment in the form of the use of the data format (JSON) as the data, so that the network topology and the cipher, and the complexity of the generated in a challenging state but also solvable. Functional validation (Table 1) and performance profiling (Figure 3) are used to provide empirical evidence of system stability under load. The "Zero Latency" performance recorded for it - less than 16 ms per frame - implies a smooth user experience, which is essential in gamified learning. A lag in generation would disrupt immersion, but as can be seen in the trace data the parallel processing architecture effectively counters this risk. The successful FSM operation confirms the existence of deterministic logic in coexistence with stochastic environments to enforce educational rules.


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

These findings support and extend previous research on procedural content generation (PCG) in education. Previous research has frequently considered PCG unpredictability as an obstacle to stringent analysis. This research stands in contrast to this opinion, proving that the incorporation of a strong validation layer, a Tool Selection FSM, can be used to constrain randomness and provide defined learning outcomes. The system is in line with the Flow Theory in game design, where time limits and the severity of punishment (through the Multiplier of Difficulty) dynamically adjust themselves to have the user in an optimum level of involvement where they do not feel frustrated by tasks that are impossible or bored by tasks that are trivial [3], [12], [17]. By automating cybersecurity scenarios, the study helps to support the move towards adaptive learning systems that respond in real time to a learner's proficiency.

There are two main implications. Technically speaking, the architecture implies the ability to control complex logic without the need of heavy server-side computation by using efficient main-thread use as can be seen in the performance trace. This reduces deployment costs and increases accessibility to students on low end devices. Pedagogically, the system solves the scaling crisis concerning cybersecurity education. Instructors normally have to write unique scenarios by hand to avoid plagiarism and rote learning. The proposed system obtains rid of this bottleneck by giving an "infinite" supply of unique missions. This is so that repeated practice can be done without content repetition, and without memorizing static answer keys, but promotes more depth of understanding of key concepts, such as how tools can be matched to the vulnerabilities [18], [19].

From a **pedagogical perspective**, the system addresses the scalability crisis in cybersecurity education. Traditionally, instructors must manually author unique scenarios to prevent plagiarism and rote memorization [20], [21]. The proposed system eliminates this bottleneck by providing a theoretical "infinite" supply of unique mission instances. This allows for repeated practice (drilling) without repetition of content, fostering deeper cognitive retention of the underlying concepts (e.g., matching tools to vulnerabilities) rather than memorizing static answer keys.

Despite the good functional performance, this study has several limitations. First, the current procedural engine only supports text-based and 2D graph topology generation, it is not yet real-time packet inspection simulation, it is also not 3D spatial environment simulation. This has limitations for those who may be advanced learners and have a control on reality. Second, though the Performance Trace did validate the responsiveness of the technical side, this study focused on functional verification (system-centric) as opposed to pedagogical efficacy (user-centric). We proved the system to work as it should, but we did not gather long term longitudinal student exam scores or retention rates.

Future research needs to focus on two important areas. First, leveraging Large Language Models (LLMs) by integrating them into the narrative generation module might enable more semantic variety by going beyond template-based descriptions and moving towards context-aware storytelling. Second, a User Experience (UX) study among a group of students is recommended to correlate the values of the Difficulty Multiplier with actual pass rates. This would allow the algorithm to be calibrated through machine learning in order to generate a personalized "Difficulty Curve", which would adapt to individual user's patterns of performance in the past.

## 4. CONCLUSION

This study successfully engineered and validated "NODE_ZERO," a browser-based tactical operations simulator designed to resolve the critical bottleneck of content exhaustion in cybersecurity education. Addressing the primary research objectives, the findings confirm that integrating a bifurcated Procedural Content Generation (PCG) algorithm with a deterministic Finite State Machine (FSM) effectively synthesizes and evaluates complex training scenarios in real time. The technical execution proved highly robust; the system concurrently randomized network topologies and cryptographic parameters while maintaining frame rendering latencies below 16 milliseconds, ensuring a seamless user experience. Furthermore, the implementation of a dynamic Difficulty Multiplier successfully scaled the operational complexity of these generated artifacts based on user inputs. This hybrid architecture proves that stochastic environment generation can be strictly governed by deterministic validation logic, creating an adaptive learning curve that accurately assesses student competency without relying on static, easily memorized answer keys.

Beyond its technical architectural contributions, this research presents substantial practical implications for both educational institutions and teaching faculties. By leveraging a lightweight Single Page Application (SPA) framework, the system empowers academic institutions to deploy infinitely variable, anti-cheat assessment environments globally, completely bypassing the financial and technical burdens of maintaining heavy server infrastructures or virtual machines. Consequently, instructors are liberated from the continuous, labor-intensive cycle of manual content authoring. This automation allows educators to redirect their valuable time and resources toward analyzing student performance metrics and providing targeted pedagogical mentoring. Ultimately, this research provides a scalable, cost-effective blueprint for the future of gamified technical training, transforming how cybersecurity competencies are drilled, assessed, and sustained in academic settings.


https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026

## REFERENCES


[1] J. R. Gómez Niño, L. P. Árias Delgado, A. Chiappe, and E. Ortega González, "Gamifying Learning with AI: A Pathway to 21st-Century Skills," *J. Res. Child. Educ.*, vol. 39, no. 4, pp. 735–750, Oct. 2025, doi: 10.1080/02568543.2024.2421974.

[2] H. Qudsi, "GAMIFICATION IN EDUCATION: BOOSTING STUDENT ENGAGEMENT AND LEARNING OUTCOMES," *ShodhKosh J. Vis. Perform. Arts*, vol. 5, no. 4, Apr. 2024, doi: 10.29121/shodhkosh.v5.i4.2024.2542.

[3] M. Farrokhi Maleki and R. Zhao, "Procedural Content Generation in Games: A Survey with Insights on Emerging LLM Integration," *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, vol. 20, no. 1, pp. 167–178, Nov. 2024, doi: 10.1609/aiide.v20i1.31877.

[4] K. Hou, J. Li, Y. Liu, S. Sun, H. Zhang, and H. Jiang, "KG-EGV: A Framework for Question Answering with Integrated Knowledge Graphs and Large Language Models," *Electronics*, vol. 13, no. 23, p. 4835, Dec. 2024, doi: 10.3390/electronics13234835.

[5] R. Sajja, Y. Sermet, M. Cikmaz, D. Cwiertny, and I. Demir, "Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education," *Information*, vol. 15, no. 10, p. 596, Sep. 2024, doi: 10.3390/info15100596.

[6] Y. Subbarayudu, G. Vijendar Reddy, M. Shankar, M. Ven, P. K. Abhilash, and A. Sehgal, "Cipher Craft: Design and Analysis of Advanced Cryptographic Techniques for Secure Communication Systems," *MATEC Web Conf.*, vol. 392, p. 01112, 2024, doi: 10.1051/matecconf/202439201112.

[7] Y. R. Kim, J. Yang, Y. Lee, and B. Earwood, "Assessing Cybersecurity Problem-Solving Skills and Creativity of Engineering Students Through Model-Eliciting Activities Using an Analytic Rubric," *IEEE Access*, vol. 12, pp. 5743–5759, 2024, doi: 10.1109/ACCESS.2023.3348554.

[8] A. N. Ghanbaripour *et al.*, "A Systematic Review of the Impact of Emerging Technologies on Student Learning, Engagement, and Employability in Built Environment Education," *Buildings*, vol. 14, no. 9, p. 2769, Sep. 2024, doi: 10.3390/buildings14092769.

[9] B. Nguyen-Viet, C. Nguyen-Duy, and B. Nguyen-Viet, "How does gamification affect learning effectiveness? The mediating roles of engagement, satisfaction, and intrinsic motivation," *Interact. Learn. Environ.*, vol. 33, no. 3, pp. 2635–2653, Mar. 2025, doi: 10.1080/10494820.2024.2414356.

[10] L. Smirani and H. Yamani, "Analysing the Impact of Gamification Techniques on Enhancing Learner Engagement, Motivation, and Knowledge Retention: A Structural Equation Modelling Approach," *Electron. J. E-Learn.*, vol. 22, no. 9, pp. 111–124, Nov. 2024, doi: 10.34190/ejel.22.9.3563.

[11] D. Kutzias and S. Von Mammen, "Recent Advances in Procedural Generation of Buildings: From Diversity to Integration," *IEEE Trans. Games*, vol. 16, no. 1, pp. 16–35, Mar. 2024, doi: 10.1109/TG.2023.3262507.

[12] A. Sarkar, M. Guzdial, S. Snodgrass, A. Summerville, T. Machado, and G. Smith, "Procedural Content Generation via Knowledge Transformation (PCG-KT)," *IEEE Trans. Games*, vol. 16, no. 1, pp. 36–50, Mar. 2024, doi: 10.1109/TG.2023.3270422.

[13] K. Krishnamurthy *et al.*, "Benefits of gamification in medical education," *Clin. Anat.*, vol. 35, no. 6, pp. 795–807, Sep. 2022, doi: 10.1002/ca.23916.

[14] V. Salauyou and W. Bułatow, "Optimized Sequential State Encoding Methods for Finite-State Machines in Field-Programmable Gate Array Implementations," *Appl. Sci.*, vol. 14, no. 13, p. 5594, Jun. 2024, doi: 10.3390/app14135594.

[15] B. M. Ampel, S. Samtani, H. Zhu, H. Chen, and J. F. Nunamaker, "Improving Threat Mitigation Through a Cybersecurity Risk Management Framework: A Computational Design Science Approach," *J. Manag. Inf. Syst.*, vol. 41, no. 1, pp. 236–265, Jan. 2024, doi: 10.1080/07421222.2023.2301178.

[16] G. Kavallieratos, G. Spathoulas, and S. Katsikas, "Cyber Risk Propagation and Optimal Selection of Cybersecurity Controls for Complex Cyberphysical Systems," *Sensors*, vol. 21, no. 5, p. 1691, Mar. 2021, doi: 10.3390/s21051691.

[17] O. Withington, M. Cook, and L. Tokarchuk, "On the Evaluation of Procedural Level Generation Systems," in *Proceedings of the 19th International Conference on the Foundations of Digital Games*, Worcester MA USA: ACM, May 2024, pp. 1–10. doi: 10.1145/3649921.3650016.

[18] Y. Deng, Z. Zeng, K. Jha, and D. Huang, "Problem- Based Cybersecurity Lab with Knowledge Graph as Guidance," *J. Artif. Intell. Technol.*, Feb. 2022, doi: 10.37965/jait.2022.0066.

[19] C. D. R. Navas Bonilla, L. M. Viñan Carrasco, J. C. Gaibor Pupiales, and D. E. Murillo Noriega, "The Future of Education: A Systematic Literature Review of Self-Directed Learning with AI," *Future Internet*, vol. 17, no. 8, p. 366, Aug. 2025, doi: 10.3390/fi17080366.

[20] J. A. Delello, W. Sung, K. Mokhtari, J. Hebert, A. Bronson, and T. De Giuseppe, "AI in the Classroom: Insights from Educators on Usage, Challenges, and Mental Health," *Educ. Sci.*, vol. 15, no. 2, p. 113, Jan. 2025, doi: 10.3390/educsci15020113.

[21] N. Martin-Alguacil, L. Avedillo, R. Mota-Blanco, and M. Gallego-Agundez, "Student-Centered Learning: Some Issues and Recommendations for Its Implementation in a Traditional Curriculum Setting in Health Sciences," *Educ. Sci.*, vol. 14, no. 11, p. 1179, Oct. 2024, doi: 10.3390/educsci14111179.



https://doi.org/10.33998/processor.2026.21.1.2777
Submitted: 06 Maret 2026; Reviewed: 31 Maret 2026; Accepted; 10 April 2026; Published: 30 April 2026