



Cognitive-Oriented Serious Game Design for Manufacturing Production-Flow Problem Solving Using Mini Drone Assembly

Handini Fatma Sari¹, Chanifah Indah Ratnasari^{2*}, Rina Mulyati³

^{1,2}Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, Indonesia

³Department of Psychology, Faculty of Psychology, Universitas Islam Indonesia, Indonesia

E-Mail: ¹handini.sari@students.uui.ac.id, ²chanifah.indah@uui.ac.id, ³rina.mulyati@uui.ac.id

Received May 12th 2026; Revised Jun 26th 2026; Accepted Jul 14th 2026; Available Online Jul 29th 2026

Corresponding Author: Chanifah Indah Ratnasari

Copyright © 2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

Manufacturing production-flow problem-solving requires Industrial Engineering students to understand how queues, work-in-process (WIP), bottlenecks, quality control, rework, and system stability interact to determine production performance. Yet conventional materials rarely allow students to observe how a single decision affects the whole system. This study aims to produce a refined, cognitive-oriented serious game design that explicitly links cognitive problem-solving, simplified production-flow rules, and game-system interaction across four progressive levels, and to evaluate it formatively as a basis for further development. Using Design-Based Research, the design was represented through Figma-based static mockups covering level progression, rule-based production behavior, feedback, and performance reports, operationalizing an Observe–Diagnose–Predict–Adjust–Stabilize–Reflect loop. Two game-design experts and five Industrial Engineering students evaluated the design through researcher-guided walkthroughs. Expert judgment produced an overall mean of 2.86 out of 5, indicating refinement needs in dynamic-system behavior and feedback. In contrast, prospective users gave 3.74 out of 5, with the strongest perception in cognitive support. Consistent with its formative objective, these results were not treated as evidence of learning effectiveness but were directly used to refine production-flow rules, feedback, and level structure, producing a four-level design artifact that provides a basis for future prototype development and effectiveness testing.

Keywords: Cognitive-Oriented Serious Game, Game Design, Manufacturing Production-Flow, Mini Drone Assembly

1. INTRODUCTION

Learning in a dynamic system requires learners to understand how one condition affects another, not only to remember separate concepts [1]. In Industrial Engineering education, this means understanding how input arrival, process time, queue, work-in-process (WIP), bottleneck, throughput, quality control, rework, and system stability connect as one system [2], [3]. It is closely related to cognitive problem-solving, since students must form mental models, predict system responses, explain cause-and-effect relations, and adapt their decisions as conditions change [4], [5], [6]. Therefore, manufacturing production-flow learning should not only help students recognize production terms but also train them to interpret problems, make decisions, and evaluate the consequences of those decisions in an interconnected production system.

Developing this competency is difficult in practice. Because production-flow behavior is dynamic and interdependent, students may grasp individual concepts yet struggle to anticipate how a single decision propagates along the line and affects the whole system. This gap between recognizing production terms and reasoning through a connected system is the main problem motivating this study, and it shows why conventional, mostly static materials cannot adequately support observation, diagnosis, prediction, adjustment, stabilization, and reflection in production-flow problem solving.

Conventional materials commonly present production-flow relationships through text, formulas, or static diagrams that explain concepts but offer limited opportunities to observe how changes in one variable affect the whole system [7]. For example, a slow station can increase WIP, increase bottleneck risk, cause throughput loss, and lead to downstream disruption. Such situations are suitable for serious game-based learning because players can interact with problems, receive feedback, and revise strategies through repeated decisions and visible consequences, reflecting decision-from-experience [8], [9].

Previous studies have shown that serious games can support engineering education and learning in manufacturing systems [10]. Virtual learning factory research shows that production system representations

can support learning interactions in Industrial Engineering education [2]. In manufacturing studies, bottlenecks, throughput, station behavior, and production-flow performance are commonly used to describe the behavior of production systems [11]. Assembly-line studies also emphasize the role of takt time and station timing in coordinating production flow [12]. However, learning content should not be limited to external instructional material; it must be embedded in the game system, interactions, feedback, and player decisions [13]. Therefore, there is still a need for a serious game design that explicitly connects cognitive problem-solving, simplified manufacturing production-flow rules, and game system interaction within a single learning artifact. Recent reviews show that digital game-based learning in engineering has grown rapidly. However, many studies still emphasize learners' perceptions, engagement, or outcomes rather than how domain-specific problem-solving is embedded in game mechanics [14]. In Industrial Engineering and manufacturing education, serious games have been developed for operations and supply chain management [15], lean manufacturing [16], and manufacturing-system configuration and analysis [10]. These studies generally focus on engagement, motivation, usability, outcomes, or functional evaluation rather than mapping production-flow problem-solving to players' cognitive processes; related work has emphasized User Interface/User Experience (UI/UX) evaluation or black-box testing of completed applications [17], [18]. From a game-based learning perspective, learning content should be integrated into the core mechanics and feedback that drive players' cognitive processing [9], [13]. To the best of our knowledge, the novelty of this study lies in integrating simplified production-flow rules, rule-based indicators, feedback, and player decisions into an Observe–Diagnose–Predict–Adjust–Stabilize–Reflect cognitive problem-solving loop across four progressive levels. In this design, production-flow indicators function as cognitive cues for reasoning, not merely as decorative or informational displays.

Based on this gap, this study proposes a cognitive-oriented serious game design for manufacturing production-flow problem-solving using a simplified mini drone assembly-line case. The design is intended to help Industrial Engineering students practice problem-solving in a production-flow context rather than to simulate real drone manufacturing. The artifact is represented through Figma-based static mockups that include prepared layouts, numerical controls, system feedback, diagnosis panels, live impact previews, simplified rule-based behavior, and performance reports. Mini drone assembly was selected as a simplified product context because drones are recognizable objects in Science, Technology, Engineering, and Mathematics (STEM) and engineering learning [19], and drone-based products can be decomposed into modular components suitable for engineering design and manufacturing activities [20]. This modularity is simplified into frame, propulsion, control-power, final assembly, inspection, Quality Control (QC), rework, and output components, letting station timing, queue, WIP, bottleneck, assembly dependency, and rework be represented across levels [12]. Therefore, mini drone assembly was used as a concrete, modular learning scenario for production-flow problem-solving, not as a simulation of real drone manufacturing.

This study uses a Design-Based Research process to develop and refine the serious game design artifact [21], [22]. The preliminary concept, level progression, Figma-based static mockups, and gameplay explanation were reviewed through expert judgment and prospective-user evaluation, and the findings were used as formative input for design refinement rather than as evidence of learning effectiveness or functional testing of a completed playable game. The objective is to produce a refined, cognitive-oriented serious game design that connects cognitive problem-solving, simplified manufacturing production-flow rules, and game-system interaction across four progressive levels: Flow and Capacity Calibration, Bottleneck and WIP Control, Line Balancing and Assembly Dependency, and QC-Rework-Based Final Production Recovery.

2. MATERIALS AND METHOD

This study used a Design-Based Research (DBR) approach to develop and refine a cognitive-oriented serious game design for manufacturing production-flow problem solving [21]. DBR was selected because it supports a design process that connects learning problems, cognitive theory, simplified production-flow concepts, game interaction, and formative evaluation [22]. The study was limited to an early-stage design and refinement cycle; it did not test learning effectiveness, apply a pre-test–post-test design, or evaluate a completed playable game application. The artifact was represented through Figma-based static mockups, gameplay explanations, simplified production-flow rules, feedback structures, and performance report designs. The DBR process consisted of three phases: analysis and exploration, design and construction, and evaluation, reflection, and design refinement, as shown in Figure 1.

2.1. Analysis and Exploration

The analysis and exploration phase defined the learning problem, the theoretical basis, the domain context, and the design requirements. The cognitive basis drew on mental models and prediction in dynamic environments [1], [4]. Causal reasoning and cognitive flexibility were reviewed as complementary processes for adapting decisions in changing situations [5], [6]. The serious game basis was reviewed through game-based learning and engineering serious game studies [9], [10]. The production-flow basis was reviewed based on the literature on bottleneck and throughput behavior in manufacturing systems [11]. Assembly-line timing

and quality-control loops were reviewed to support takt time, station timing, QC, and rework representation [3], [12].

Contextual input was collected in an approximately 40-minute offline discussion with five eighth-semester Industrial Engineering students from Universitas Islam Indonesia on 19 March 2026, which was used as preliminary design input for the production-flow learning context. It covered topics the students had studied, including manufacturing systems, production planning and control, bottlenecks, throughput, line balancing, WIP, QC, and simulation tools such as FlexSim. The same five students later joined the prospective-user evaluation after the mockup walkthrough. Thus, the discussion served as preliminary design input, and the questionnaire responses served as prospective user evaluation data.

This phase produced the design requirements: the Observe, Diagnose, Predict, Adjust, Stabilize, and Reflect gameplay loop as the cognitive requirement, and production-flow requirements covering input arrival, conveyor transfer, station processing, queue or WIP buffer, bottleneck risk, blocking, starvation, throughput, QC queue, rework, and system stability. Together, they defined the game as a cognitive problem-solving environment for production-flow learning, not a full industrial simulator.

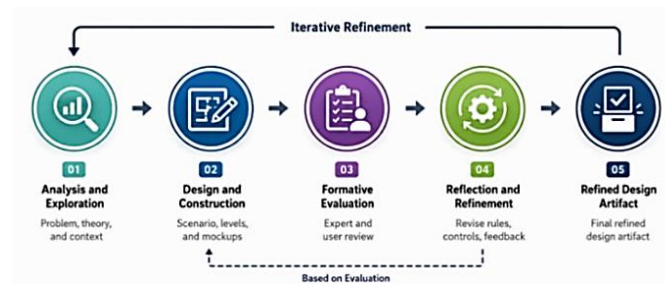


Figure 1. Research Methodology

2.2. Design and Construction

The design and construction phase translated the design requirements into a serious game design artifact. The mini drone assembly scenario was selected because it provides a concrete and modular production context that can be divided into station-based and takt-related production-flow problems [12]. The scenario was used as a simplified manufacturing problem environment, not as a full industrial drone-manufacturing simulation.

Depending on the level, the production-flow components were source, conveyor, processing station, transfer queue, WIP buffer, assembly or inspection point, QC/rework area, packaging, warehouse, and output buffer, with items moving from input arrival through conveyor transfer, station processing, queue or buffer movement, inspection or assembly, rework when required, packaging, and final output.

The design has four progressive levels: Flow and Capacity Calibration, Bottleneck and WIP Control, Line Balancing and Assembly Dependency, and QC-Rework-Based Final Production Recovery. Each provides a prepared layout; the player does not build it but observes the running system, adjusts numerical parameters, and monitors the response.

Depending on the level, the player-adjustable parameters included input arrival interval, conveyor transfer time, station process time, buffer release interval, merge ratio, function test time, rework release interval, and packaging process time. Queues and WIP rose when input rate, process time, or downstream capacity became unbalanced, reflecting simplified bottleneck and throughput behavior [11]. Bottleneck risk was indicated when a station had excessive process time, high load, or an accumulated upstream queue [11]. QC and rework behaviors were represented through the QC queue, rework buffer, and retest flow, and their effects on throughput and system stability [3].

The game rules and threshold values were positioned as a simplified rule-based, conceptual representation that specifies feedback behavior, warning status, win/fail and collapse conditions, and report popups; they were not automatic outputs of a Unity-based engine. Target outputs were 20 frame modules (Level 1), 25 drive-frame modules (Level 2), 30 assembled mini drone units (Level 3), and 35 finished mini drones (Level 4). Thresholds were set for stable, warning, failed, and collapse states based on indicators such as system stability, queue capacity, WIP buffer, bottleneck risk, final assembly idle time, line balance score, QC queue, rework buffer, retest queue, and QC pass rate.

The materials were prepared in Figma as static UI mockups representing the gameplay screen, station layout, numerical controls, diagnosis panel, warning indicators, live impact preview, and performance report. They were not clickable prototypes, a Unity-based application, or an interactive simulator.

Generative AI tools were used to assist in creating production-asset visuals for the gameplay figures, including conveyors, machines, workstations, and related production-line elements, and in refining manuscript language. The Figma layouts, production-flow logic, metrics, rules, level designs, data analysis, interpretation,

and final decisions were developed, reviewed, and verified by the authors. AI was not used to determine the research findings, and the authors take full responsibility for the article.

2.3. Formative Evaluation

The formative evaluation followed the DBR orientation of using evidence to improve an educational design artifact [21]. The evaluated materials included the game concept, level progression, Figma-based static mockups, gameplay explanation, simplified production-flow rules, feedback structure, and questionnaire forms. Because the artifact was not a playable prototype, evaluators reviewed the design through the researcher's explanation and visual walkthrough.

Expert judgment was provided by two lecturers with expertise in game design and development. Each expert attended a 30-minute Zoom session and completed a Google Form containing 21 Likert-type statements and 4 open-ended questions. The instrument covered game interaction, UI clarity, dynamic representation of the production system, feedback mechanisms, cognitive and decision-making support, and development feasibility.

The prospective-user evaluation involved five eighth-semester Industrial Engineering students from Universitas Islam Indonesia who had studied manufacturing systems and related concepts, including production flow, bottlenecks, throughput, queues, WIP, quality control, capacity, and manufacturing-system simulation. In a face-to-face session, students provided contextual input, reviewed the game concept and Figma mockups through a guided explanation, and completed a Google Form with 18 Likert-type statements and 6 open-ended questions covering domain and learning relevance, UI and operational-flow clarity, and perceived cognitive support.

The questionnaires used a five-point Likert scale (1 = strongly disagree to 5 = strongly agree). The evaluation domains captured educational-game design aspects, including interaction clarity, feedback quality, perceived cognitive support, and development feasibility. Expert items examined, for example, whether the production-system behavior was understandable and whether feedback helped identify production-flow problems, while prospective-user items examined whether the interface clarified production flow and encouraged prediction before adjustment and cause-and-effect reasoning.

2.4. Data Analysis, Reflection, and Design Refinement

Likert-type responses were analyzed descriptively using frequency distribution, median, mean, standard deviation, and positive response percentage [23]. Frequency distributions and medians were used to describe response tendencies, while means and standard deviations were used as supporting indicators for comparing evaluation domains. The item mean, domain mean, standard deviation, and positive response percentage were calculated using Equations (1)–(4). The item mean was calculated using Equation (1).

$$\bar{x}_j = \frac{\sum_{i=1}^n x_{ij}}{n} \quad (1)$$

where $\bar{x}_j$ is the mean score of the item j , x_{ij} is the score given by the respondent i to item j and n is the number of respondents. The domain mean was calculated using Equation (2).

$$\bar{x}_D = \frac{\sum_{j=1}^k \sum_{i=1}^n x_{ij}}{n \times k} \quad (2)$$

where $\bar{x}_D$ is the mean score of an evaluation domain, x_{ij} is the score given by the respondent i to item j , n is the number of respondents, and k is the number of items within the domain. The standard deviation was calculated using Equation (3).

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (3)$$

where SD is the standard deviation, x_i is an individual score, $\bar{x}$ is the mean score, and n is the number of responses. The positive response percentage was calculated using Equation (4).

$$PRP = \frac{f_4 + f_5}{N} \times 100\% \quad (4)$$

where PRP is the positive response percentage, f_4 and f_5 are the number of responses with scores of 4 and 5, respectively, and N is the total number of responses in the item or domain.

Open-ended responses were analyzed through thematic summarization. The comments were coded based on relevant design issues and grouped into refinement themes [24]. The themes included production

context clarity, dynamic system behavior, interaction clarity, feedback quality, user interface readability, game-like learning experience, and cognitive support.

The findings were synthesized into design-refinement priorities by comparing the problem found, the evidence source, the design response, and the refined feature, guiding refinement of the production-flow rules, numerical controls, feedback mechanisms, level structure, interface readability, and the performance report. The refined artifact, therefore, represents the outcome of evaluation, reflection, and refinement, not evidence of learning effectiveness or functional performance of a completed playable game.

3. RESULTS AND DISCUSSION

3.1. Analysis and Exploration Results

The analysis and exploration phase identified the design basis: learning focus, production-flow context, cognitive gameplay loop, production indicators, and feedback requirements. These elements were centered on cognitive problem-solving in production-flow situations, particularly how students observe, diagnose, predict, adjust, stabilize, and reflect on system behavior.

The contextual discussion indicated that production-flow learning is commonly associated with layout interpretation, station timing, bottleneck analysis, WIP behavior, QC, simulation practice, and output evaluation. It also indicated that students needed to see where a problem occurs, what causes it, and how it affects downstream flow. These inputs were translated into prepared layouts, visible routes, timing parameters, WIP and queue indicators, bottleneck warnings, selected-station diagnosis, live impact preview, and report-based reflection.

Accordingly, the resulting design adopted a prepared production-flow layout rather than a layout-building approach. In this design, players observe the running system, diagnose production-flow problems, predict the effects of parameter changes, adjust numerical controls, stabilize the system, and reflect on their decisions through a performance report. This structure formed the basis for the refined design artifact.

3.2. Design Construction Results

The design construction stage produced a refined serious game design artifact comprising gameplay flow, a storyboard, four-level progression, Figma-based static mockups, simplified production-flow rules, feedback components, and performance report design. The mini drone assembly scenario served as a simplified problem environment rather than a full industrial drone manufacturing simulation.

The gameplay has three phases. The Briefing Phase introduces the mission, product context, learning focus, target output, initial issue, and prepared layout. The Live Control Phase lets players observe production indicators, diagnose problems, predict the effect of parameter changes, adjust numerical controls, and stabilize the system. The Review Phase presents a performance report that connects player decisions to production-flow consequences. The overall flow is shown in Figure 2.

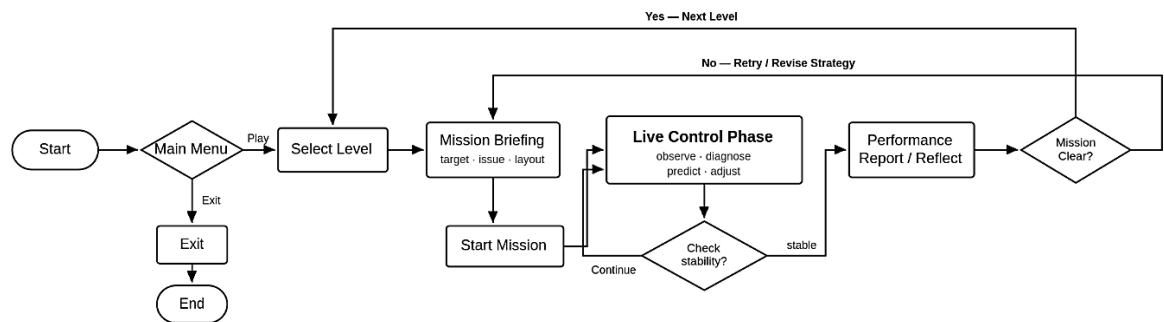


Figure 2. Refined gameplay flowchart

Figure 2 illustrates the gameplay flow. It starts with level selection and mission briefing, followed by the Live Control Phase, where players observe the system, diagnose problems, preview parameter impacts, and adjust controls. During this phase, stability and mission progress are continuously checked. At the end, players receive a performance report and either proceed to the next level or retry with a revised strategy. The storyboard in Figure 3 summarizes the main interaction sequence of the proposed serious game design.

Figure 3 presents the storyboard sequence, including the main menu, level selection, mission briefing, Live Control Phase, report or feedback display, and game-over condition. This sequence represents the Observe, Diagnose, Predict, Adjust, Stabilize, and Reflect loop within the proposed serious game design.

The final design consists of four progressive levels, each with a distinct production-flow problem, player task, adjustable parameter set, system feedback, success indicator, and cognitive emphasis, as summarized in Table 1.

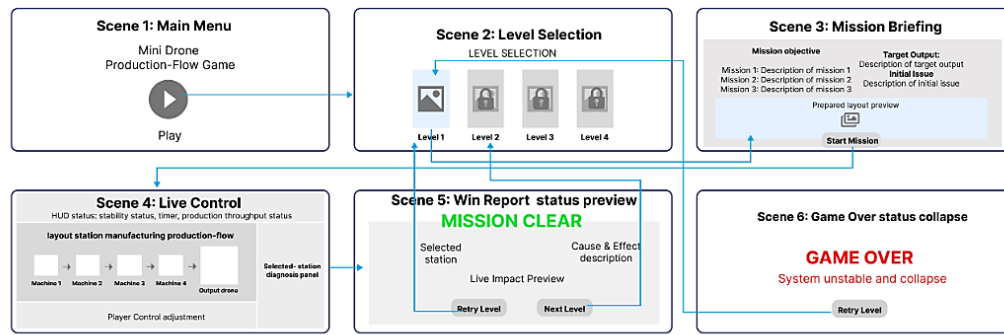


Figure 3. Storyboard

Table 1. Refined Level Structure

Level	Player task	Adjustable parameter	System feedback and success indicator	Main Cognitive emphasis
Level 1: Flow and Capacity Calibration	Balance input arrival, transfer, and station processing to maintain a stable flow	Input arrival interval, conveyor transfer time, Machine 1 process time, Machine 2 process time	Queue level, station load, throughput, takt-time comparison, and system stability; success if output target and stability are achieved	Mental model formation of flow, capacity, and stability
Level 2: Bottleneck and WIP Control	Identify bottleneck risk and control WIP in a serial production line	Input arrival interval, WIP Buffer A release interval, Machine 2 process time, and Machine 3 process time	WIP level, bottleneck risk, blocking, starvation, station imbalance, and throughput stability; success if WIP and output remain stable	Causal reasoning among station capacity, WIP, and output
Level 3: Line Balancing and Assembly Dependency	Synchronize parallel module flows before final assembly	Kitting release interval, Drive Module Line process time, Control-Power Line process time, Final Assembly process time	Module buffer level, final assembly idle time, line balance score, assembly dependency, and final output; success if module supply is balanced	Predictive reasoning and cognitive flexibility across parallel flows
Level 4: QC-Rework-Based Final Production Recovery	Recover the final production flow under QC and rework pressure	Assembled unit arrival interval, Function Test time, Rework release interval, Packaging process time	QC queue, rework buffer, retest queue, packaging readiness, final output, and system stability; success if final output recovers and bottleneck risk is controlled	Cognitive flexibility and systems thinking in recovery conditions

The parameters exposed to the player represent the core levers of a production-flow system rather than every variable of a real factory. Each level isolates a small parameter set mapped to a recognized production-flow mechanism and a target cognitive process: input arrival, transfer, and station process times govern input pacing and station capacity against takt (Level 1, mental model formation); the buffer-release interval and station process times govern work-in-process and bottleneck capacity (Level 2, causal reasoning); the kitting-release interval and parallel module-line cycle times govern module synchronization and assembly dependency (Level 3, predictive reasoning); and function-test time, rework-release interval, and packaging time govern quality-inspection capacity and rework flow (Level 4, cognitive flexibility and systems thinking). Jointly covering input pacing, capacity, work-in-process, synchronization, and quality-rework, these parameters are sufficient to pose the intended production-flow problems while keeping each level cognitively focused. Variables outside this scope, such as machine breakdowns, operator scheduling, and stochastic variability, were deliberately excluded to preserve clear cause-and-effect reasoning and are noted as future extensions.

The refined UI mockup uses Level 3 because it shows line balancing and assembly dependencies within a parallel production flow. The UI displays module buffer levels, final assembly idle time, line balance score, warning status, selected-station diagnosis, live impact preview, and numerical controls. The performance report is also shown for Level 3 to maintain consistency in the gameplay problem and reflection (Figures 4 and 5).

Figure 4 shows the player observing parallel module lines, identifying buffer imbalance, predicting parameter effects, adjusting parameters, and monitoring the final-assembly response. Figure 5 shows the report summarizing mission results, output, system stability, module balance, assembly idle time, station performance, and learning points, supporting reflection by linking player actions to production-flow consequences.



Figure 4. Refined Gameplay UI for Level 3: Line Balancing and Assembly Dependency



Figure 5. Level 3 Production Performance Report

3.3. Formative Evaluation Results

The preliminary materials were evaluated through expert judgment and prospective-user evaluation via researcher-guided walkthroughs; the results were therefore interpreted as formative design evidence, not evidence of learning effectiveness or playable-prototype performance. The evaluator profile and evaluation criteria are summarized in Table 2.

Table 2. Evaluator and Evaluation Criteria

Evaluation domain	Code	Number of participants	Selection criteria	Evaluation focus	Instrument
Game design expert	E1-E2	2	Lecturers with a background in game design, game development, or serious games	Game interaction, UI clarity, dynamic production-system representation, feedback mechanism, cognitive support, decision-making support, and development feasibility	21 Likert-type statements and 4 open-ended questions
Prospective users	PU1-PU5	5	Eighth-semester Industrial Engineering students who had studied manufacturing systems and production-flow concepts	Domain and learning relevance, UI and operational-flow clarity, and perceived cognitive support	18 Likert-type statements and 6 open-ended questions

The expert judgment results are summarized in Table 3. The overall mean was 2.86 (median 3.00, SD 1.39, positive response 45.24%). The lowest domain was the feedback mechanism (mean 2.17, positive response 16.67%), and the dynamic production-system representation also needed improvement (mean 2.63).

These results indicate that the preliminary design needed clearer production-flow behavior, decision consequences, and rule-based feedback.

Table 3. Expert Judgment Result Summary

Evaluation domain	Number of items	Mean	Median	SD	Positive Response (%)	Interpretation	Refinement implication
UI and interaction clarity	4	3.38	3.50	0.74	50.00	Moderate	Clarify the interaction model.
Level progression	3	3.00	4.00	1.55	66.67	Moderate	Clarify the level of complexity.
Dynamic production-system representation	4	2.63	2.50	1.77	50.00	Needs refinement	Define production-flow rules
Feedback mechanism	3	2.17	2.00	1.33	16.67	Low	Strengthen feedback response
Cognitive support and decision-making	7	2.93	3.00	1.44	42.86	Needs refinement	Embed cognitive support
Overall expert judgment	21	2.86	3.00	1.39	45.24	Needs Refinement	Refine design artifact

A closer reading of Table 3 reveals a consistent pattern rather than uniformly low scores. The two weakest domains, feedback mechanism (2.17) and dynamic production-system representation (2.63), are exactly the aspects that depend on runtime behavior and interaction, which static Figma mockups cannot fully convey: the experts could read the screens but could not watch the system respond to a decision. In contrast, UI and interaction clarity scored highest (3.38), indicating that the layout and information structure were already legible in a static form. The high standard deviation (1.39) across only two experts is consistent with this split. Taken together, the results point to a design whose production-flow concept is sound but whose system behavior and feedback were not yet observable, an outcome expected at this static, pre-prototype stage that directly motivated the refinements in Table 5.

The prospective user results are summarized in Table 4. The overall mean was 3.74 (SD 0.79; median 4.00; positive response rate 66.67%). The highest domain was cognitive support (mean 3.92; positive response rate 76.00%), suggesting that the design was perceived as supporting prediction, decision-making, cause-and-effect reasoning, and systems thinking. However, domain and learning relevance (3.60) indicate that product identity, station roles, and production routes still need clearer representation.

Table 4. Prospective-User Evaluation Result Summary

Evaluation domain	Number of items	Mean	SD	Median	Positive Response (%)	Interpretation
Domain and learning relevance	7	3.60	0.77	4.00	54.29	Moderate positive tendency
UI and operational-flow clarity	6	3.77	0.82	4.00	73.33	Positive tendency
Cognitive support	5	3.92	0.76	4.00	76.00	Positive tendency
Overall prospective-user evaluation	18	3.74	0.79	4.00	66.67	Moderate positive tendency

The prospective users responded more positively than the experts, and the contrast is informative in itself. Students rated cognitive support highest (3.92), and their overall mean (3.74) exceeded the expert mean (2.86) because the groups evaluated the design through different lenses. Students judged it against their own coursework, assessing whether the production-flow problems of bottleneck, WIP, line balancing, and QC rework matched concepts they had studied and could support prediction and decision-making. In contrast, experts judged its completeness as an interactive game system. The lower domain and learning relevance score (3.60) indicates that the product identity and production route are still perceived as generic, echoing the experts' call for a more explicit representation of the production flow. Both groups thus converged on the same gap, a relevant cognitive concept that still needed clearer representation and feedback, strengthening confidence that the Table 5 refinement priorities reflect a real design need rather than a single perspective.

Figure 6 summarizes the mean scores for each evaluation domain across both groups. It makes the contrast visible at a glance: the experts' lowest ratings fall on feedback mechanism and dynamic production-system representation, whereas the prospective users' highest rating is on cognitive support, and the prospective-user means sit above the expert means across comparable domains.

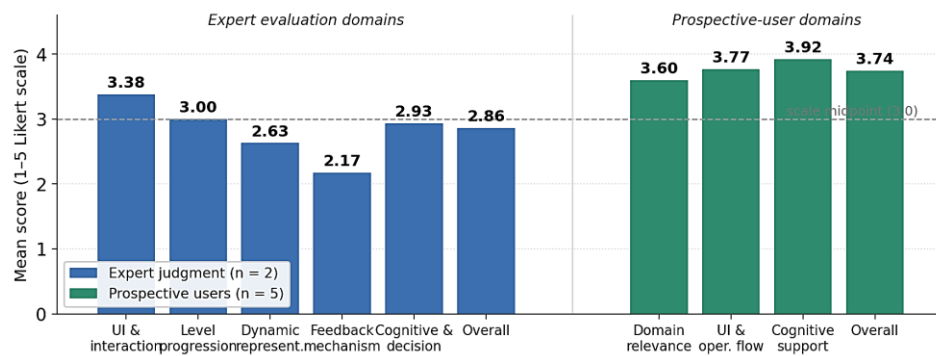


Figure 6. Expert and prospective-user mean scores by evaluation domain

3.4. Evaluation-Based Design Refinement

The evaluation findings were translated into design refinement actions, consistent with the DBR emphasis on using contextual evidence to improve design artifacts [21], [22]. Quantitative results identified weak domains, while open-ended responses and contextual input explained the issues behind the scores. The refinement mapping is shown in Table 5.

Table 5. Evaluation-Based Design Refinement Synthesis

Finding	Source and evidence	Before the refinement/design issue	Refined feature
The feedback mechanism was unclear.	Expert judgment; feedback mechanism mean 2.17, and a positive response 16.67%	Feedback did not clearly explain the consequences of the decision or why the system became unstable.	Added diagnosis panel, warning indicators, live impact preview, cause-effect information, and production performance report
The dynamic production logic was not explicit enough	Expert judgment and comments; dynamic production-system representation mean 2.63	Queue, WIP, bottleneck, blocking, starvation, QC queue, and rework behavior were not sufficiently represented	Defined rule-based production behavior for arrival, process time, queue, WIP, bottleneck, blocking, starvation, QC queue, rework, throughput, and stability
Player control and system response were not sufficiently linked	Expert feedback on decision support and interaction clarity	Player controls were perceived as abstract and did not clearly show measurable production effects	Revised controls into numerical production parameters according to the level of focus
Product context and production route were still generic	Prospective-user evaluation; domain relevance mean 3.60 and user comments on route clarity	Product identity, station roles, conveyor direction, and production areas were still generic	Specified mini drone module progression, station labels, conveyor routes, WIP buffers, QC/rework areas, warehouse, and output buffer
Cognitive support needed stronger operational embedding.	Expert and prospective-user synthesis; prospective-user cognitive support mean 3.92	Cognitive goals were present but not fully embedded into production-flow indicators and feedback.	Linked queue, WIP, bottleneck risk, station load, throughput, QC queue, rework load, and stability to Observe, Diagnose, Predict, Adjust, Stabilize, and Reflect
Production-flow context needed a concrete representation.	Contextual discussion on layout, station timing, bottleneck, QC, FlexSim, and output analysis	The initial context was still too general as a manufacturing-themed game	Used prepared layout, station diagnosis, timing parameters, representation of flow behavior over time, and report-based reflection

Table 5 shows that refinement extended beyond visual improvement. The results led to clearer feedback mechanisms, more explicit production-flow rules, numerical control parameters, more specific mini drone assembly routes, and stronger alignment between production indicators and cognitive problem-solving. The final artifact is therefore a refined serious game design, not a completed playable game.

3.5. Discussion

The formative evaluation indicates that the preliminary design had a relevant learning direction but required a clearer representation as a serious game system. Experts identified feedback mechanisms and

dynamic production-system representation as the main weaknesses, showing that the design needed to clarify not only screen elements but also system behavior and the effects of player decisions. Prospective users responded more positively, especially regarding cognitive support for prediction, decision-making, cause-and-effect reasoning, and systems thinking, but still noted that product identity, station roles, conveyor routes, and operational flow needed to be more explicit. Therefore, the refinement focused on making the design readable as a production-flow problem-solving environment rather than merely an industrial-themed interface.

The main contribution of this study is a refined serious game design artifact that embeds problem-solving for the manufacturing production flow into gameplay rather than presenting it solely as external instructional material [13]. Production indicators act as representational cues that support observation, diagnosis, prediction, adjustment, stabilization, and reflection, connecting cognitive processes with simplified production-flow behavior through player decisions, system feedback, and report-based reflection.

Previous educational game studies have focused on different evaluation approaches. Hasanah et al. [17] focused on the UI/UX design of a 2D educational game and evaluated the prototype using SUS and UEQ. Aprilianto et al. [18] focused on Android-based educational game development and evaluated it through black-box testing. In contrast, this study focuses on the formative refinement of a cognitive-oriented serious game design for production-flow problem-solving. Its contribution is therefore not UI/UX usability or functional testing of a completed game, but a design-refinement study that integrates cognitive problem-solving, simplified production-flow rules, game interaction, and formative evaluation into a single artifact.

These findings can be interpreted in relation to previous serious game and game-based learning studies. Prior serious games in engineering and manufacturing education have shown the potential of game-based approaches for manufacturing-system configuration, operations and supply-chain learning, and lean manufacturing education [10], [15], [16]. Related educational-game studies have also emphasized UI/UX evaluation [17] and black-box functional testing of completed game applications [18]. In contrast, this study focuses on the formative refinement of a cognitive-oriented serious game design artifact, where production-flow indicators are not treated only as visual information but as cognitive cues embedded in game mechanics and feedback. This is consistent with game-based learning theory, which emphasizes that learning content should be integrated into the mechanics and feedback that guide players' cognitive processing [9], [13]. Therefore, the contrasting evaluation results indicate that the cognitive intent of the design was visible to prospective users, but the feedback loop and dynamic-system representation still needed clearer operationalization in the static mockups. This explains why the refinement focused on the diagnosis panel, live impact preview, and rule-based behavior rather than cosmetic visual changes.

These findings are formative evidence for improving the design. With two experts and five prospective users, they guide refinement rather than support statistical generalization. Future work should implement the refined design as a playable Unity prototype, test it through direct user interaction, and evaluate its effect on students' cognitive problem-solving outcomes with a larger group.

4. CONCLUSION

This study produced a refined, cognitive-oriented serious game design for manufacturing production-flow problem-solving using a simplified mini drone assembly scenario. Cognitive problem-solving is the main learning orientation, while production flow serves as the problem environment. Mental model formation, causal reasoning, predictive reasoning, and cognitive flexibility are operationalized through the Observe, Diagnose, Predict, Adjust, Stabilize, and Reflect loop. The artifact was represented through Figma-based static UI mockups, gameplay flow, storyboard, four-level progression, simplified rule-based production behavior, feedback components, and performance report design, with the four levels connecting cognitive processes to production-flow indicators such as queue, WIP, bottleneck risk, throughput, QC queue, rework, line balance, and system stability.

The novelty and main contribution of this study lie in the integration of cognitive problem-solving processes, simplified production-flow rules, and game-system interaction within a single serious game design artifact. Rather than positioning production-flow indicators only as informational displays, this study designs simplified, rule-based indicators as cognitive cues that are explicitly mapped to the Observe, Diagnose, Predict, Adjust, Stabilize, and Reflect problem-solving loop across four progressive levels. The artifact was formatively evaluated and refined as a serious game design rather than assessed as a completed, playable application, a usability product, or a functionally tested game. This directly addresses the objective of producing a refined design that connects cognitive problem-solving, simplified production-flow rules, and game-system interaction.

The formative evaluation showed that the preliminary design was relevant but still required refinement. Expert judgment yielded an overall mean of 2.86 out of 5, indicating a need to clarify dynamic-system behavior, feedback, interaction logic, and decision consequences. In contrast, prospective-user evaluation yielded 3.74 out of 5, a moderately positive perception, especially in cognitive support. These findings guided refinement of the production-flow rules, numerical controls, diagnosis panel, warning indicators, live impact preview, level structure, layout readability, and performance report. The study is limited to a refined design artifact and

formative evaluation using Figma-based static UI mockups; it did not measure learning effectiveness, usability, or the functional performance of a completed playable game. Future research should implement the design as a playable Unity prototype, test it through direct user interaction, and evaluate its effect on students' cognitive problem-solving outcomes with a larger group.

ACKNOWLEDGMENTS

During the preparation of this manuscript, the authors used generative AI tools only to assist in creating production-asset visuals for the gameplay figures and refining the manuscript language. AI was not used to conduct data analysis, interpret evaluation results, or determine the research findings. The interface layouts were arranged in Figma, while the production-flow logic, metrics, rules, and level designs were specified by the authors. The authors reviewed and verified all figures and text and take full responsibility for the article.

REFERENCES

- [1] S. N. Groesser and M. Schaffernicht, "Mental models of dynamic systems: taking stock and looking ahead," *Syst. Dyn. Rev.*, vol. 28, no. 1, pp. 46–68, Jan. 2012, doi: 10.1002/sdr.476.
- [2] W. Terkaj, M. Urgo, P. Kovács, E. Tóth, and M. Mondellini, "A framework for virtual learning in industrial engineering education: development of a reconfigurable virtual learning factory application," *Virtual Real.*, vol. 28, no. 3, p. 148, Aug. 2024, doi: 10.1007/s10055-024-01042-8.
- [3] M. C. Magnanini, O. Demir, M. Colledani, and T. Tolio, "Performance evaluation of multi-stage manufacturing systems operating under feedback and feedforward quality control loops," *CIRP Annals*, vol. 73, no. 1, pp. 349–352, Jan. 2024, doi: 10.1016/j.cirp.2024.04.015.
- [4] R. Tikhonov and S. DeDeo, "Prediction, Explanation, and Control: The Use of Mental Models in Dynamic Environments," *Open Mind*, vol. 7, pp. 894–916, Nov. 2023, doi: 10.1162/opmi_a_00112.
- [5] L. Y. Lee *et al.*, "Cognitive flexibility training for impact in real-world settings," *Curr. Opin. Behav. Sci.*, vol. 59, p. 101413, Oct. 2024, doi: 10.1016/j.cobeha.2024.101413.
- [6] M. K. Goddu and A. Gopnik, "The development of human causal learning and reasoning," *Nature Reviews Psychology*, vol. 3, no. 5, pp. 319–339, Apr. 2024, doi: 10.1038/s44159-024-00300-5.
- [7] E. Bauer, N. Heitzmann, and F. Fischer, "Simulation-based learning in higher education and professional training: Approximations of practice through representational scaffolding," *Studies in Educational Evaluation*, vol. 75, p. 101213, Dec. 2022, doi: 10.1016/j.stueduc.2022.101213.
- [8] S. Olschewski, A. Luckman, A. Mason, E. A. Ludvig, and E. Konstantinidis, "The Future of Decisions From Experience: Connecting Real-World Decision Problems to Cognitive Processes," *Perspectives on Psychological Science*, vol. 19, no. 1, pp. 82–102, Jan. 2024, doi: 10.1177/17456916231179138.
- [9] J. L. Plass, B. D. Homer, and C. K. Kinzer, "Foundations of Game-Based Learning," *Educ. Psychol.*, vol. 50, no. 4, pp. 258–283, Oct. 2015, doi: 10.1080/00461520.2015.1122533.
- [10] M. Urgo, W. Terkaj, M. Mondellini, and G. Colombo, "Design of serious games in engineering education: An application to the configuration and analysis of manufacturing systems," *CIRP J. Manuf. Sci. Technol.*, vol. 36, pp. 172–184, Jan. 2022, doi: 10.1016/j.cirpj.2021.11.006.
- [11] A. Skoogh, M. Thürrer, M. Subramaniyan, A. Matta, and C. Roser, "Throughput bottleneck detection in manufacturing: a systematic review of the literature on methods and operationalization modes," *Prod. Manuf. Res.*, vol. 11, no. 1, Dec. 2023, doi: 10.1080/21693277.2023.2283031.
- [12] A. Huchzermeier and T. Mönch, "Mixed-model assembly lines with variable takt and open stations," *Prod. Oper. Manag.*, vol. 32, no. 3, pp. 704–722, Mar. 2023, doi: 10.1111/poms.13893.
- [13] N. Szilas, "Understanding Learning Content Integration in Games," *International Journal of Game-Based Learning*, vol. 15, no. 1, pp. 1–25, Sep. 2025, doi: 10.4018/IJGBL.389730.
- [14] C. Udeozor, R. Toyoda, F. Russo Abegão, and J. Glassey, "Digital games in engineering education: systematic review and future trends," *European Journal of Engineering Education*, vol. 48, no. 2, pp. 321–339, Mar. 2023, doi: 10.1080/03043797.2022.2093168.
- [15] G. Romagnoli, M. Galli, D. Mezzogori, and D. Reverberi, "A Cooperative and Competitive Serious Game for Operations and Supply Chain Management," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 18, no. 15, pp. 17–30, Dec. 2022, doi: 10.3991/ijoe.v18i15.35089.
- [16] M. Gianpiero, P. Pedrazzoli, G. Landolfi, F. Daniele, and E. Montini, "Introducing Active Learning and Serious Game in Engineering Education: 'Experience from Lean Manufacturing Course,'" in *Advances in Production Management Systems. Production Management Systems for Responsible Manufacturing, Service, and Logistics Futures*, vol. 690, Cham, 2023, pp. 363–377. doi: 10.1007/978-3-031-43666-6_25.
- [17] F. U. Hasanah, R. W. Satrio, M. I. Hanafri, and S. Maisaroh, "Implementasi Design Thinking pada UI/UX Game Edukasi 2D Hafalan Hadist Menggunakan System Usability Scale (SUS) dan User Experience Questionnaire (UEQ)," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 711–721, Apr. 2025, doi: 10.57152/malcom.v5i2.1969.

-
- [18] M. A. J. Aprilianto, E. Wijayanti, and A. A. Chamid, "Game Adventure of Cakra Versi Cerita Rakyat Indonesia Sebagai Media Pembelajaran Interaktif," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 1, pp. 414–422, Jan. 2025, doi: 10.57152/malcom.v5i1.1879.
 - [19] M. M. Bolick, E. A. Mikhailova, and C. J. Post, "Teaching Innovation in STEM Education Using an Unmanned Aerial Vehicle (UAV)," *Educ. Sci. (Basel)*, vol. 12, no. 3, p. 224, Mar. 2022, doi: 10.3390/educsci12030224.
 - [20] F. Akasheh *et al.*, "Additive Manufacturing-Enabled Modular Drone Design Development by Multidisciplinary Engineering Student Team," in *2022 ASEE Annual Conference & Exposition*, Minneapolis, MN: ASEE Conferences, Aug. 2022. doi: 10.18260/1-2--42122.
 - [21] C. Hoadley and F. C. Campos, "Design-based research: What it is and why it matters to studying online learning," *Educ. Psychol.*, vol. 57, no. 3, pp. 207–220, Jul. 2022, doi: 10.1080/00461520.2022.2079128.
 - [22] S. McKenney and T. C. Reeves, "Educational design research for relevant & robust scholarship," *J. Comput. High. Educ.*, vol. 37, no. 2, pp. 614–638, Jun. 2025, doi: 10.1007/s12528-025-09456-2.
 - [23] G. M. Sullivan and A. R. Artino, "Analyzing and Interpreting Data From Likert-Type Scales," *J. Grad. Med. Educ.*, vol. 5, no. 4, pp. 541–542, Dec. 2013, doi: 10.4300/JGME-5-4-18.
 - [24] S. K. Ahmed *et al.*, "Using thematic analysis in qualitative research," *Journal of Medicine, Surgery, and Public Health*, vol. 6, p. 100198, Aug. 2025, doi: 10.1016/j.glmedi.2025.100198.