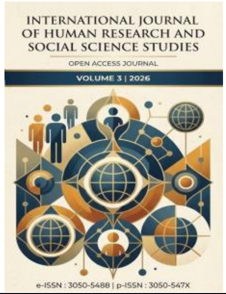
International Journal of Human Research and Social Science Studies

ISSN(p): 3050-547X, ISSN(e): 3050-5488

Volume 03 Issue 06 June, 2026

DOI: https://doi.org/10.55677/ijhrsss/15-2026-Vol03I06

Page No : 647-661



Prompted Props, Human Pipelines: Evaluating AI-Generated 3D Assets for Game-Ready Environments

Andrew Begemann[1], James Hutson[2]

[1]Department of Game Design, Lindenwood University

[2]Department of Art History, AI, and Visual Culture, Lindenwood University

ABSTRACT: Generative AI systems increasingly promise rapid 3D asset production for game development, yet their practical viability depends on whether generated models can move beyond visual preview into editable, optimized, engine-ready workflows. This article presents a practice-led comparative case study of a stylized fantasy tavern environment produced through two workflows: a human-authored Blender pipeline and an AI-assisted pipeline using Meshy 6 and Hunyuan 3D. Using a fixed asset list, shared visual theme, documented prompts, production-time tracking, visual comparison, topology inspection, UV-map analysis, and post-generation labor accounting, the study evaluates whether text-to-3D tools function as production substitutes, ideation accelerators, or conditional asset sources. Results indicate that AI-assisted generation substantially reduced first-pass production time: the Hunyuan-assisted reconstruction required 238 minutes compared with 716.06 minutes for the human-authored scene. However, the apparent time savings were accompanied by substantial technical debt, including dense triangulated geometry, fragmented UV maps, inconsistent prompt adherence, material-editing constraints, clipping during placement, and loss of texture integrity during attempted decimation. The human-authored workflow required more labor at the modeling stage but produced assets with greater intentionality, editability, scale control, and technical legibility. The findings support a human-in-the-loop model in which generative AI contributes most effectively to ideation, rough prop exploration, and early prototyping, while artists and technical artists remain necessary for optimization, art direction, retopology, UV reconstruction, material refinement, and engine validation.

Corresponding Author:
Andrew Begemann

Published Online:
June 30, 2026

KEYWORDS:
generative AI, text-to-3D, game asset production, 3D modeling, human-AI collaboration, game-ready assets, production pipeline

Cite the Article: *Begemann, A., Hutson, J. (2026). Prompted Props, Human Pipelines: Evaluating AI-Generated 3D Assets for Game-Ready Environments. International Journal of Human Research and Social Science Studies,3(6),647-661.* https://doi.org/10.55677/ijhrsss/15-2026-Vol03I06

1. INTRODUCTION

Generative AI has moved from speculative discourse into the everyday workflows of game studios, independent developers, students, and tool vendors, but the shift has not resolved the practical question most relevant to 3D production: whether AI-generated assets can survive the transition from a visually impressive preview to a deployable, editable, and optimized real-time model. The pressure point is visible across the industry (Nagpal, Chugh, & Rohilla, 2026; Ternar, et al., 2026; Werning, 2024). The 2026 State of the Game Industry report indicates that more than one-third of surveyed game professionals were using generative AI as part of their work, while a majority believed that the technology was having a negative impact on the game industry, with especially unfavorable views among visual artists, technical artists, designers, narrative workers, and programmers (Game Developers Conference, 2026). This tension reflects a deeper production contradiction. Asset generators promise speed, abundance, and lower barriers to entry, but game development requires more than visual output; it requires assets that obey polygon budgets, preserve stylistic coherence, accept revision, support materials, scale predictably, import cleanly into engines, and remain legally and ethically defensible (Fukaya, Daylamani-Zad, & Agius, 2026; Savery, 2026). A generated tavern chair, mug, fireplace, or room shell may appear functional in a browser viewer, yet it may still be unsuitable for production if its mesh is too dense, its UV space is unreadable, its topology cannot deform or subdivide, or its texture cannot be edited without rebuilding the asset. This article addresses that gap by reframing AI asset generation as a pipeline problem rather than an image-quality problem. It asks not whether a generator can produce a medieval-looking prop, but whether that prop can become part of a game-ready environment without reintroducing substantial labor downstream.

The study analyzed here began as a comparative game-design project centered on a fantasy tavern environment created in two versions: one human-authored in Blender and one reconstructed through AI-generated assets produced with Meshy 6 and Hunyuan 3D. The project is particularly useful as a case study because it does not limit evaluation to impressionistic judgments about whether AI output looks attractive. Instead, it records production time, prompt results, unwanted model features, download constraints, UV-map legibility, triangle counts, object counts, decimation behavior, and placement labor. Those categories allow the examination what might be called asset-readiness: the combined visual, technical, and procedural qualities that determine whether a 3D object can be adopted into a game pipeline (Costa, 2026). Asset-readiness differs from surface appeal because it foregrounds conditions that are often invisible to non-specialists, including topology flow, quad versus triangle structure, material-slot organization, retopology burden, texture-editing possibility, collision requirements, level-of-detail planning, and engine performance (Ramos, 2017). The article therefore treats AI generation as a partial workflow intervention, which may compress ideation and first-pass object creation, but it may also transfer labor from initial modeling to selection, correction, optimization, and integration. The central argument here is that AI does not eliminate asset-production labor; it relocates labor into later and more technical stages of the pipeline.

This framing matters because scholarship on generative AI in game design often emphasizes creativity, prototyping, workflow acceleration, or developer perception, while the asset-level technical consequences of text-to-3D generation remain comparatively underdeveloped. Alharthi (2025), for example, found that game designers and developers recognized AI's capacity to accelerate ideation, prototyping, and repetitive tasks, yet also identified concerns about dependency, originality, and the weakening of human-authored content. French et al. (2023) similarly examined OpenAI-supported game-development education through case studies that demonstrate how generative tools can assist creative exploration while also requiring critical oversight. Begemann and Hutson (2024) documented generative AI across a broader creative pipeline, including concept art, 3D model generation, and implementation challenges, concluding that AI shows promise but still faces limitations when assets must be integrated into game environments. The present article builds on that trajectory by narrowing the inquiry to a specific production problem: the game-readiness of AI-generated 3D environmental assets. The fantasy tavern case offers a bounded design space, a fixed object list, and direct comparison between human-authored and AI-assisted workflows. As a result, the article contributes a practical benchmark for evaluating generative AI not as a generalized creative force but as a production tool whose viability depends on the technical and artistic labor that remains after prompting.

2. LITERATURE REVIEW

Generative AI research in game development has increasingly moved away from abstract speculation toward the situated study of tools, users, and workflows. Traditional artificial intelligence in games has long concerned non-player character behavior, player modeling, search, procedural content generation, and adaptive systems; generative AI adds a new layer by enabling designers to request text, code, images, textures, models, animations, and dialogue through natural-language prompting (Yannakakis & Togelius, 2018). In this newer context, AI is often promoted as a way to reduce production bottlenecks and democratize asset creation, particularly for students and independent developers without large production teams. However, the literature also suggests that the benefit of AI depends on the kind of work being measured. Alharthi (2025) showed that developers perceive AI as useful for ideation and prototyping, but they also worry that it can constrain innovation if designers over-rely on model outputs or allow generated content to dominate art direction. Moon et al. (2024) similarly described generative AI in educational game design as a field of nuanced challenges, in which students may benefit from rapid experimentation but still need design literacy, ethical awareness, and technical judgment. These studies are valuable because they reject a simple productivity narrative. They suggest that generative AI must be understood as a sociotechnical system that reconfigures creative work, skill development, and production responsibility.

Human-AI co-creation research provides an especially useful conceptual vocabulary for this study because text-to-3D asset production is neither fully automated nor purely manual. Lin et al. (2023) argue that mixed-initiative co-creativity systems should not be reduced to one-directional prompting in which the user merely specifies a desired output and the system returns a finished artifact. Stronger forms of co-creation involve iteration, scrutability, explanation, adaptation, and shared control. This distinction is directly relevant to game assets. A text-to-3D system may generate a table or a chair, but a production workflow must still ask whether the model's geometry can be inspected, modified, repaired, rigged, decimated, baked, retextured, or used as a collision object. The user needs not only a model but an accountable object whose structure can be understood and changed. Creativity research complicates the issue further. Holzner et al. (2025) found that AI may improve human creative performance in some settings while reducing diversity in human-AI collaboration, which indicates that efficiency and originality may not move in the same direction. Wadinambiarachchi et al. (2024) likewise found that generative image support can produce design fixation, depending on how users prompt and respond to AI outputs. For game production, these findings suggest that AI can accelerate the search for possible assets but can also narrow design thinking if artists accept outputs too quickly or allow generator defaults to set the aesthetic grammar of the environment.

The problem becomes more technically specific in the domain of 3D assets. A game-ready prop is not simply a rendered image wrapped around a geometric surface; it is a structured object designed for real-time performance, material control, scale consistency, and possible interaction. Kuusela (2022) describes a game asset pipeline through concepting, modeling, texturing, UV mapping, retopology, and map baking, which together establish the difference between an appealing model and an efficient production asset. Text-to-3D systems typically compress the early phases of that pipeline by producing geometry and texture from text or image prompts, but they do not necessarily supply the same intentional mesh structure that a modeler creates through controlled topology. Hunyuan3D 2.0, for instance, is described as a large-scale 3D synthesis system with separate shape and texture components, namely Hunyuan3D-DiT and Hunyuan3D-Paint (Zhao et al., 2025). Hunyuan3D 2.5 continues this trajectory through higher-fidelity and more detailed textured asset generation, while still presenting itself as a system for producing 3D assets through learned synthesis rather than manual topology design (Lai et al., 2025). Other recent text-to-3D systems, such as 3DTopia-XL and Magic123, similarly foreground the problem of generating detailed and textured 3D objects from textual or image conditions, yet their research emphasis is often fidelity, alignment, and representation rather than the full range of post-generation game-production tasks (Chen et al., 2024; Qian et al., 2023). Consequently, asset-level studies remain necessary because production viability depends on how generated geometry behaves after export, import, inspection, and optimization.

Tool vendors increasingly acknowledge these production needs, even as marketing language can blur the distinction between generation and readiness. Meshy describes itself as a text-to-3D and image-to-3D platform that can support game development, PBR textures, remeshing, export workflows, and rapid iteration (Meshy LLC, n.d.). Such claims are important because they show that commercial tools are not merely positioning themselves as novelty generators; they are entering the vocabulary of asset pipelines, game engines, and production budgets. Yet the existence of export options, PBR map support, or remesh settings does not guarantee that every generated model will satisfy the standards of a particular game project. An asset may export to GLB or FBX while still exceeding a scene's triangle budget, fragmenting UV islands in ways that make hand editing difficult, or including unwanted semantic features that a prompt did not request. This article therefore treats vendor claims as part of the context but not as evidence of production success. The comparative case examines what happened when publicly accessible AI tools were used to recreate a bounded tavern asset set. It asks whether the practical affordances described by tools are reflected in the resulting models once those models are placed in a coherent environment, evaluated against topology and UV expectations, and compared to human-authored assets built with game-production considerations in mind.

A further strand of scholarship concerns authorship, disclosure, and the ethics of AI-generated game content. Mikalonyte and Kneer (2022) showed that folk intuitions about whether AI can make art remain complex, with perceptions of agency, intention, and artistic status shaping how people evaluate AI-generated works. In game development, those questions intersect with intellectual-property uncertainty, platform governance, and labor displacement. Steamworks documentation now requires developers to disclose generative AI use in a product's content survey, distinguishing pre-generated AI content produced during development from live-generated content produced while a game is running; for live-generated systems, developers must also describe guardrails intended to prevent illegal output (Valve, n.d.). This requirement matters for the present study because AI-assisted assets are not just technical artifacts. They become part of a documentation ecology in which developers may need to record tool use, assess licensing conditions, explain content provenance, and decide whether AI-generated assets align with marketing claims. The article does not attempt to resolve copyright doctrine, but it recognizes that game-ready viability includes legal and procedural readiness alongside geometry and texture. A production asset must be technically usable, aesthetically coherent, and compatible with disclosure obligations.

3. METHODOLOGY

3.1 Research Questions

The article is organized around four research questions. RQ1 asks how AI-assisted production compares with human-only production in time-on-task for a stylized fantasy game environment. This question uses the project log's human-production time, Meshy generation time, Hunyuan generation time, and Hunyuan placement time to determine whether AI actually reduces labor when scene assembly is included rather than measuring only the browser-generation stage. RQ2 asks how accurately text-to-3D systems follow prompts for a coherent fantasy tavern environment. This question considers prompt adherence at the asset level, including successful outputs, failed or partial outputs, and unwanted additions such as extra doors, sinks, bottles, fireplaces, or room structures. RQ3 asks what technical barriers appear when generated assets are evaluated for real-time game use, including topology, polygon density, UV layout, texture editability, and decimation behavior. This question is central to the article because the difference between visual plausibility and game-readiness becomes most visible when models are inspected structurally. RQ4 asks what forms of human labor remain necessary after AI generation and how those tasks should be incorporated into a viable human-AI asset-production workflow. Together, the questions move the discussion from general AI adoption to a more precise evaluation of where prompting helps, where it fails, and where human expertise remains indispensable **(Table 1).**

Table 1. Comparative Workflow Design

Study component	Human-authored Blender workflow	AI-assisted Meshy/Hunyuan workflow
Scene target	Stylized fantasy tavern with 40 by 28 foot common room and functional prop layout	Reconstruction of the same general tavern environment using generated props and comparable placement
Asset scope	Room shell, tables, chairs, bar, shelves, barrels, trophy heads, doors, fireplaces, tableware, lanterns	Prompts for the same asset categories, followed by selection of usable generated variants
Primary labor	Modeling, measuring, UV marking, procedural materials, duplication, scene population	Prompting, variant selection, download/export checks, scaling, duplication, alignment, clipping correction
Evaluation emphasis	Time, visual coherence, mesh control, UV/editability, scene completeness	Time, prompt adherence, visual output, topology, UV maps, polygon density, placement labor

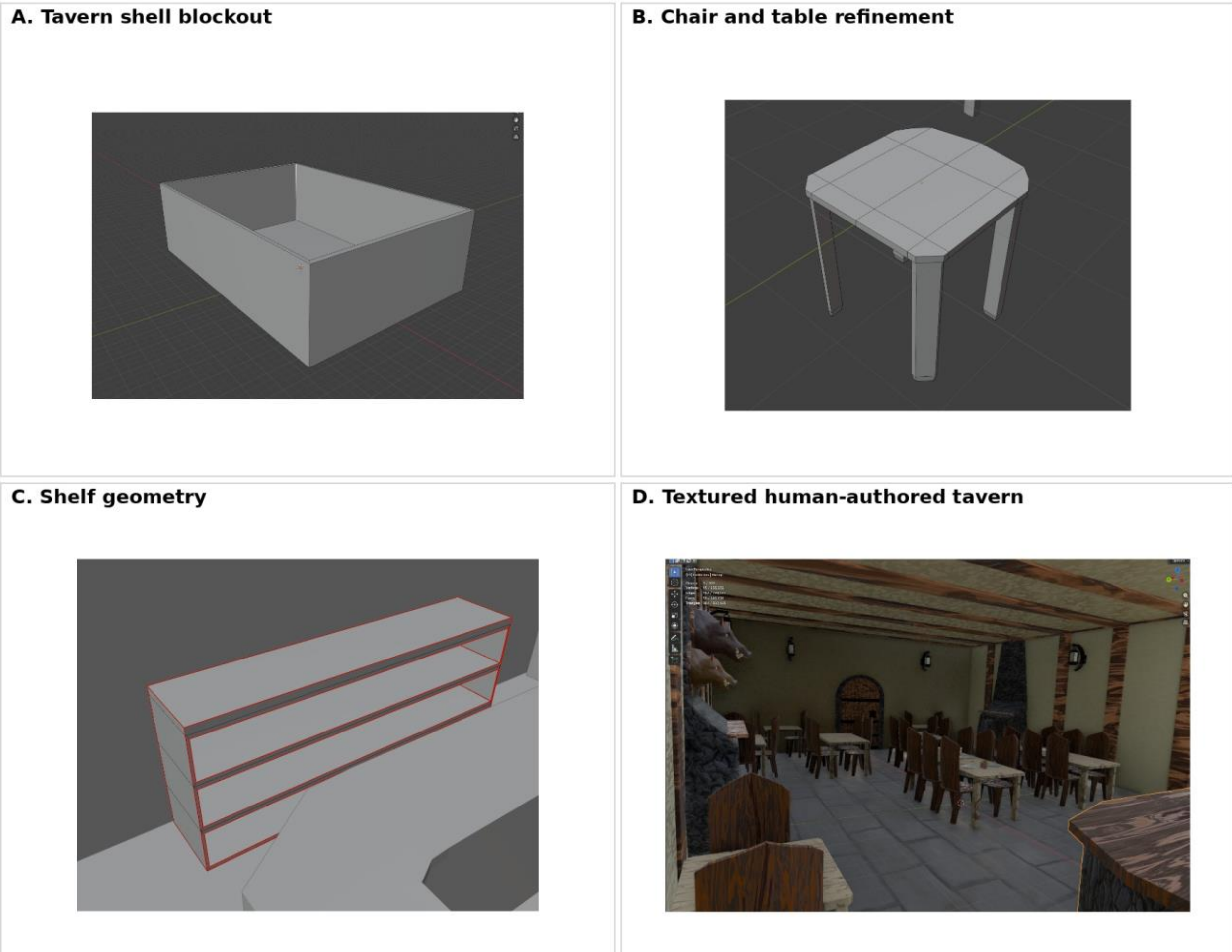


Figure 1. Human-authored modeling and texturing workflow for the tavern environment

This study uses a practice-led comparative case-study design. The object of comparison is a fantasy tavern environment produced in two versions: a human-authored Blender scene and an AI-assisted reconstruction using generated assets from Meshy 6 and Hunyuan 3D. The setting was selected because a tavern is a familiar role-playing game environment that requires several common environmental props: architectural shell, tables, chairs, bar, shelves, barrels, doors, fireplaces, mugs, plates, goblets, shot glasses, wall lanterns, and decorative trophies **(Figure 1)**. The genre also imposes style expectations associated with medieval fantasy, stylized wood and stone materials, functional interior layout, and believable object scale. The human workflow created the room and assets manually in Blender through polygonal modeling, duplication, UV marking, procedural materials, and scene placement. The AI workflow attempted to recreate the same general asset set using prompts entered into Meshy 6 and Hunyuan 3D, followed by import and placement in Blender to approximate the human-authored room. The study therefore compares not a single object but a small production ecology, which is important because game environments rarely consist of isolated props. Their viability depends on how objects work together at scene scale.

The human-authored workflow served as the baseline. The tavern shell began as a 40 by 28-foot room with walls extruded from an inset plane, followed by tables, chairs, bar elements, shelves, barrels, boar head trophies, doors, fireplaces, drinkware, lanterns, and procedural materials. The project log records both modeling and texturing time, including tutorial-supported material creation and later scene population. This baseline is not presented as an ideal professional studio workflow, since the original project was produced by a single artist and included experimentation, measurement, corrections, and tutorial consultation. Its value is that the author controlled the geometry, object proportions, UV seams, materials, duplicates, and layout through direct manipulation in Blender. Consequently, the human workflow provides a meaningful comparison for technical intentionality. It shows how asset decisions are made during modeling rather than after generation. Mesh density, shape readability, material assignment, and UV layout are incorporated into the act of building the model, even when the asset takes longer to produce.

The AI-assisted workflow began with Meshy 6, which was used to generate the same broad class of assets through text prompts or image-supported generation. The first prompt attempted to produce a 40 by 28-foot medieval room with wooden beams and plaster surfaces, but the output introduced unrequested doors, sconces, and a fireplace, and did not follow the specified room dimensions. A second prompt produced only a rectangular plank, so the room shell was not obtained in a usable form through Meshy. The tool nevertheless generated a range of tavern props, including tables, chairs, plates, mugs, shot glasses, goblets, fireplaces, doors, shelves, boar trophies, bar forms, wall lanterns, and barrels. Meshy outputs were visually acceptable in several cases, but the process revealed two categories of limitation: prompt drift, in which the model included features not requested, and access friction, in which downloads for Meshy 5 and 6 outputs required a Pro subscription during the project. Because the objective was to compare practical asset creation under accessible conditions, the paywall changed the role of Meshy in the study. Meshy remained useful for observing generation behavior and UV-map quality, but the final AI-assisted scene was assembled from Hunyuan 3D assets that could be downloaded.

Hunyuan 3D was then used to generate and download assets through prompts adapted from the more successful Meshy attempts. Hunyuan's workflow produced four models per prompt in many cases and included textures with the generated geometry, allowing the researcher to choose among variants. Prompts included phrases such as 'a rectangular wooden medieval table,' 'a wooden medieval chair,' 'a medieval pewter plate,' 'a medieval wooden beer mug,' 'a glass shot glass,' 'a medieval drinking goblet,' 'a stone fireplace and chimney with a wooden shelf and stone hearth extension,' 'a wooden tavern door with stone framing, metal cross bars, and a metal handle,' 'a wide, empty wooden bookshelf,' 'a boar head trophy on a wooden display plaque,' 'a wooden L shaped bar with a stone countertop,' 'Hanging Wall Lantern,' and 'Large wooden beer barrel with spigot.' Hunyuan outputs were not treated as final by default. They were imported into Blender, scaled, duplicated, aligned, and placed to approximate the human-authored scene. This scene-assembly stage is methodologically important because it captures labor often omitted from demonstrations of AI tools. A model that appears instantly generated still requires spatial decision-making, proportional adjustment, collision avoidance, and stylistic evaluation when inserted into a game environment.

Evaluation used seven criteria: production time, prompt adherence, visual fidelity, technical readiness, UV-map coherence, texture editability, and post-generation correction burden. Production time included the recorded human-only total, Meshy generation duration, Hunyuan generation duration, and Hunyuan placement duration. Prompt adherence considered whether the generated asset matched requested object type, scale, structure, and semantic features. Visual fidelity assessed surface appeal, stylistic coherence, and whether an object plausibly belonged in a fantasy tavern. Technical readiness examined object count, triangle count, mesh density, topology type, and evidence that the asset could be optimized for real-time use. UV-map coherence addressed whether the texture space appeared legible enough for targeted manual editing. Texture editability considered whether colors, materials, and surface properties could be revised without extensive reconstruction. Post-generation correction burden included scaling, clipping, duplication, variant selection, retopology implications, decimation problems, and the need for secondary low-poly meshes. The figures embedded in this article were selected from the original project documentation to illustrate the article's analytic claims rather than to reproduce the full thesis archive. They show the human workflow, representative Meshy outputs and map fragmentation, Hunyuan output variation, placement labor, final room comparison, and decimation failure.

4. RESULTS

The time comparison reveals the most visible advantage of AI-assisted production, but it also shows why generation time alone is an incomplete metric **(Table 2)**. The human-authored Blender scene required 716.06 minutes, or 11 hours, 56 minutes, and 4 seconds, including modeling, procedural material creation, application, and population of the tavern. Meshy 6 produced the needed models and textures in 117 minutes, though the outputs could not be used as the final downloadable asset set without a paid subscription during the project. Hunyuan 3D generated the asset set in 79 minutes, and the subsequent placement and adjustment of downloaded models in Blender required 159 minutes. The Hunyuan-assisted reconstruction therefore required 238 minutes in total, or 3 hours and 58 minutes, which is 478.06 minutes less than the human-only workflow and approximately a 66.8% reduction in logged production time. At a surface level, this result strongly supports the value of AI for first-pass generation. However, the time comparison also shows that the largest practical gain occurs before technical optimization is considered. The Hunyuan total includes asset generation and placement, but it does not include full retopology, UV reconstruction, texture baking, collision setup, LOD creation, or engine-performance validation. The speed advantage is real, but it measures a prototype-complete scene rather than a fully optimized game-ready scene.

Table 2. Production-Time Comparison by Workflow Stage

Workflow stage	Logged time	Interpretive note
Human-authored Blender scene	716.06 minutes, or 11 hours, 56 minutes, 4 seconds	Includes modeling, procedural materials, texturing, duplication, and room population
Meshy 6 generation	117 minutes, or 1 hour, 57 minutes	Produced asset previews and textures, but download access for tested outputs required a paid tier
Hunyuan 3D generation	79 minutes	Generated downloadable assets and multiple variants for most prompts
Hunyuan placement in Blender	159 minutes, or 2 hours, 39 minutes	Includes import, scaling, alignment, duplication, clipping correction, and shelf/table population
Hunyuan-assisted total	238 minutes, or 3 hours, 58 minutes	Approximately 66.8% less logged time than the human-authored scene, excluding full retopology and engine validation

Prompt adherence varied substantially by tool, prompt, and object type. Meshy had difficulty generating the room shell with the specified dimensions and constraints **(Figure 2).** The first room prompt produced a square room-like space with unrequested doors, sconces, and a central fireplace, while the second prompt produced a rectangular plank rather than a usable room. This failure is notable because architectural shell generation requires spatial relationships that are more constrained than small prop generation. Meshy performed better on isolated props, especially chairs, pewter plates, goblets, fireplaces, boar trophies, wall lanterns, and barrels, but it also introduced unwanted features such as extra decorative surfaces, unrequested bottles, possible sinks, holes, or mismatched material suggestions. Hunyuan followed several objects prompts more successfully and offered multiple variants per prompt, which supported choice and art-direction filtering. However, Hunyuan also exhibited prompt limitations: shelf prompts generated bookcases with books until the word 'empty' was added, wall-lantern prompts generated pillars before being revised, and room prompts produced erratic objects or unusable shell variants. These examples suggest that prompt engineering functions as production labor. It is not a negligible linguistic preface to asset creation; it is an iterative design task that determines what assets become available for selection and how much downstream correction will be needed.

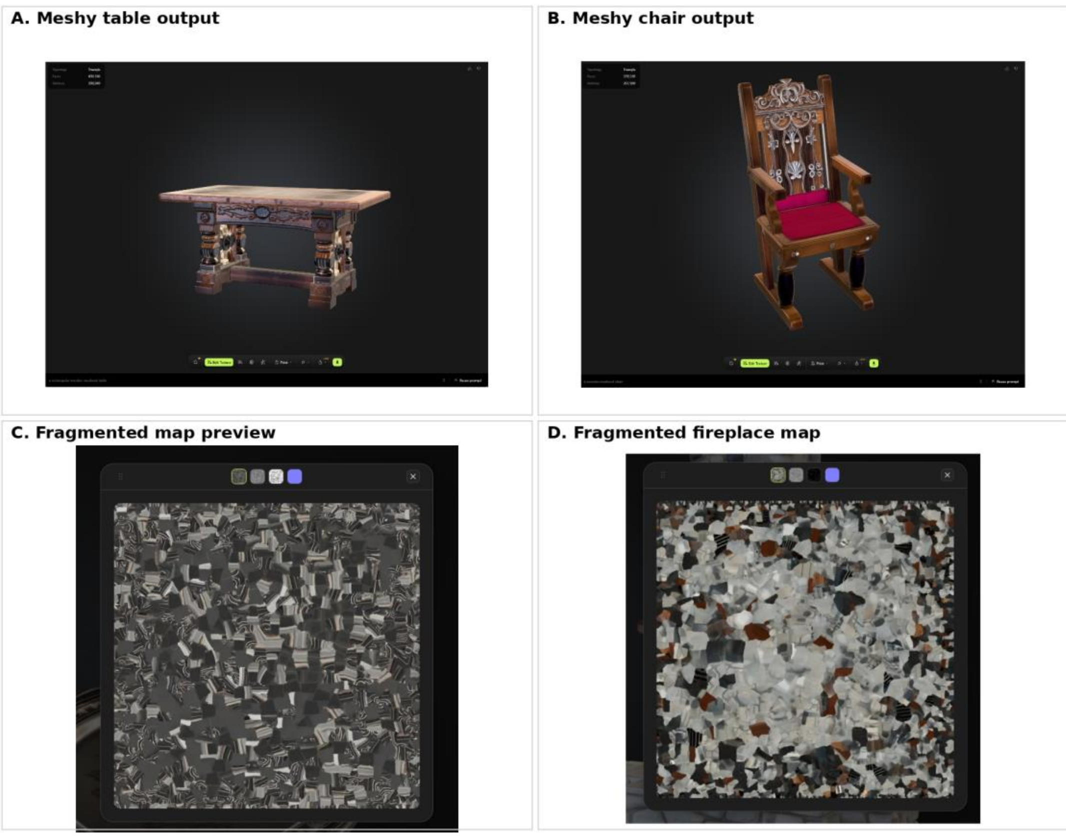


Figure 2. Representative Meshy 6 outputs and fragmented map previews

Visual fidelity favored the AI-generated Hunyuan assets at first glance in several cases **(Figure 3).** The selected table, chair, door, fireplace, barrel, plate, mug, and bar assets had ornamental details, stylized textures, and a polished appearance that exceeded what a novice or intermediate modeler might produce quickly. Hunyuan's outputs were especially strong when the prompt requested a familiar object category with a rich visual tradition, such as a medieval chair or stone fireplace. Meshy also generated visually compelling outputs, including ornate furniture and a strong boar trophy, although some maps and textures showed inconsistent material assignments. The human-authored room, by contrast, had more modest surface complexity but stronger internal control. Materials were procedural and could be adjusted, duplicated, recolored, or replaced; geometry could be selected and edited; and proportions could be changed in relation to the intended layout. This contrast reveals a key distinction between apparent polish and editable control. AI assets may look more detailed in static screenshots, but their detail is often entangled with dense geometry and baked or fragmented texture information. The human workflow produces less automatic ornamentation but more intentionality, and intentionality becomes valuable whenever a designer needs to revise a prop to match a changing level, gameplay requirement, art direction note, or performance budget.

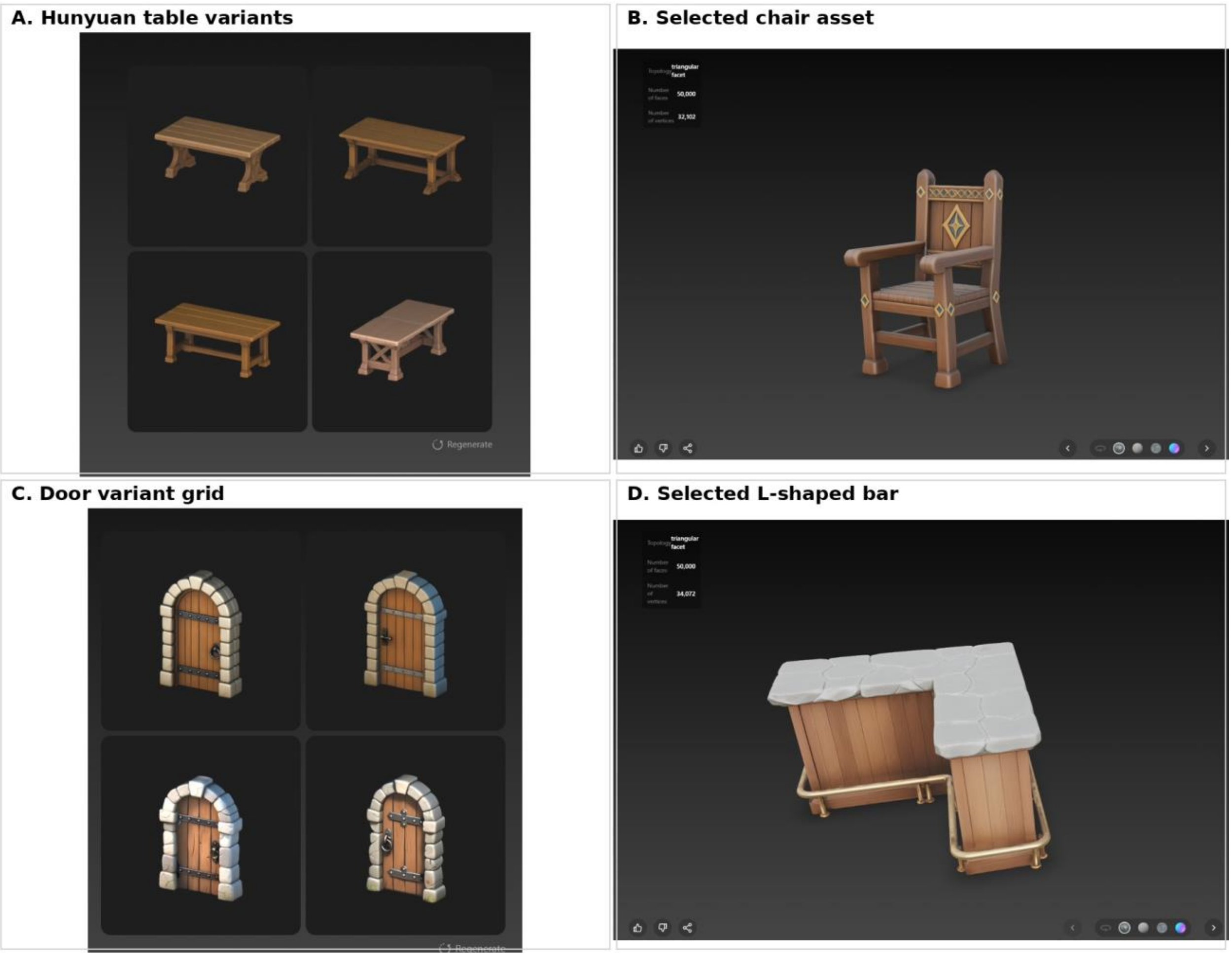


Figure 3. Representative Hunyuan 3D variant generation and selected assets

The technical-readiness comparison is the strongest evidence against treating text-to-3D tools as direct replacements for human asset creation. The human-authored scene contained 123 objects, 141,008 faces, and 293,828 triangles. The Hunyuan 3D scene contained 103 objects and 32,781,505 triangles. Although the AI-assisted scene had fewer objects, it had roughly 111.6 times as many triangles. This difference is not merely a numerical curiosity. Triangle count affects memory, rendering cost, batching, collision complexity, file size, editability, and the feasibility of using multiple such assets in a real-time game level. Hunyuan did not provide the same level of quad-based topology control as the human Blender workflow, and generated assets were largely triangulated. Triangulation is not inherently unusable for static props, but dense triangulated topology becomes problematic when assets must be animated, optimized, deformed, decimated, or edited. The project attempted to decimate a generated boar's head trophy in Blender, but the model began losing triangles, producing holes and damaging texture alignment **(Figure 4)**. This result demonstrates that automatic polygon reduction may not be sufficient when the generated mesh and texture map are not structured for controlled simplification **(Table 3)**. A more viable production path would require a secondary low-poly model, retopology, projection or baking from the high-resolution generated model, and revised texture management.

Table 3. Technical-Readiness Matrix

Criterion	Human-authored room	AI-assisted Hunyuan room	Production implication
Objects	123 objects	103 objects	Fewer objects did not reduce technical weight in the AI-assisted scene
Geometry complexity	141,008 faces; 293,828 triangles	32,781,505 triangles	AI scene was roughly 111.6 times heavier by triangle count
Topology control	Artist-controlled modeling decisions and UV seams	Dense triangulated outputs with limited quad control	Generated assets require inspection before production adoption
UV/editability	Intentional unwrapping and material control	Fragmented texture maps and difficult targeted editing	AI textures can look good while resisting revision
Optimization path	Geometry decisions built into modeling process	Decimation damaged boar trophy geometry and texture alignment	Retopology and baking likely required for production use

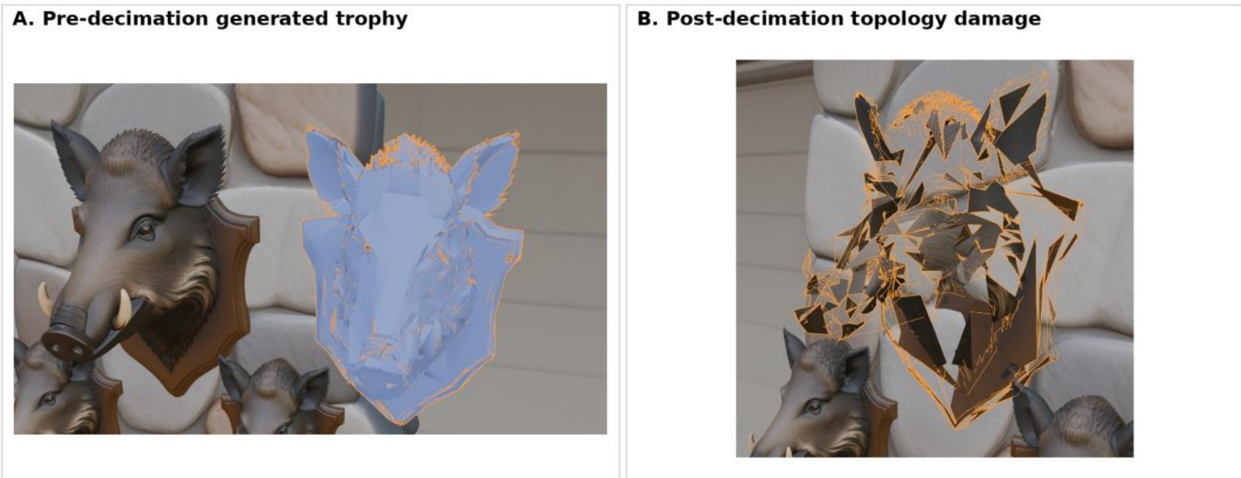


Figure 4. Decimation and topology damage in an AI-generated boar trophy

UV-map coherence further differentiates the workflows. In the human-authored scene, seams were marked during modeling so that UV unwrapping and material assignment could be planned in relation to each object. The resulting UVs were not necessarily presented as professional final maps, but they reflected intentional unfolding and could support material revision. Meshy and Hunyuan outputs, by contrast, produced texture maps that appeared fragmented across many small faces or islands, making it difficult to infer which part of the map corresponded to which visible model surface. The project documentation describes these AI-generated UV maps as lacking cohesion, with polygons scattered across the texture space rather than arranged in a way that would allow easy manual editing. This is a critical production limitation because texture editability is not optional in many game projects. Art directors may request color changes, weathering variations, decal placement, damage states, faction variants, or localization-sensitive details. If UV space is unreadable, every targeted texture change becomes slower and more error-prone. AI-generated texture maps can produce convincing first-pass results, but they may reduce the asset's long-term usefulness if their structure resists human revision.

Post-generation labor was most visible during scene assembly. Hunyuan assets had to be imported, scaled, duplicated, aligned, and adjusted against the human-authored reference scene. The bar required duplication, rotation, partial deletion, and adaptation to approximate the original layout, and the attempt to fill geometry was made difficult by mesh density. Tables and chairs needed size adjustment, but the Hunyuan table's understructure and the chair's armrests prevented the chairs from sliding under the table in the same way as the human-authored scene **(Figure 5).** Doors had to be shifted to avoid clipping with generated room pillars, which then required adjustments to shelves and barrels. Tableware had to be placed with tables hidden and then moved upward to rest on the Hunyuan tabletops. Shelf objects required array modifiers and additional alignment work to avoid clipping. These details matter because they show that AI generation did not produce a complete room from a prompt. It produced a set of attractive but independent objects that required human spatial reasoning to become a coherent environment. The AI-assisted scene was much faster than the human-only scene, but the remaining human labor involved the same types of situated judgment that level designers and environment artists routinely perform: scale, proportion, collision, camera readability, object relationships, and stylistic unity **(Figure 6)**.

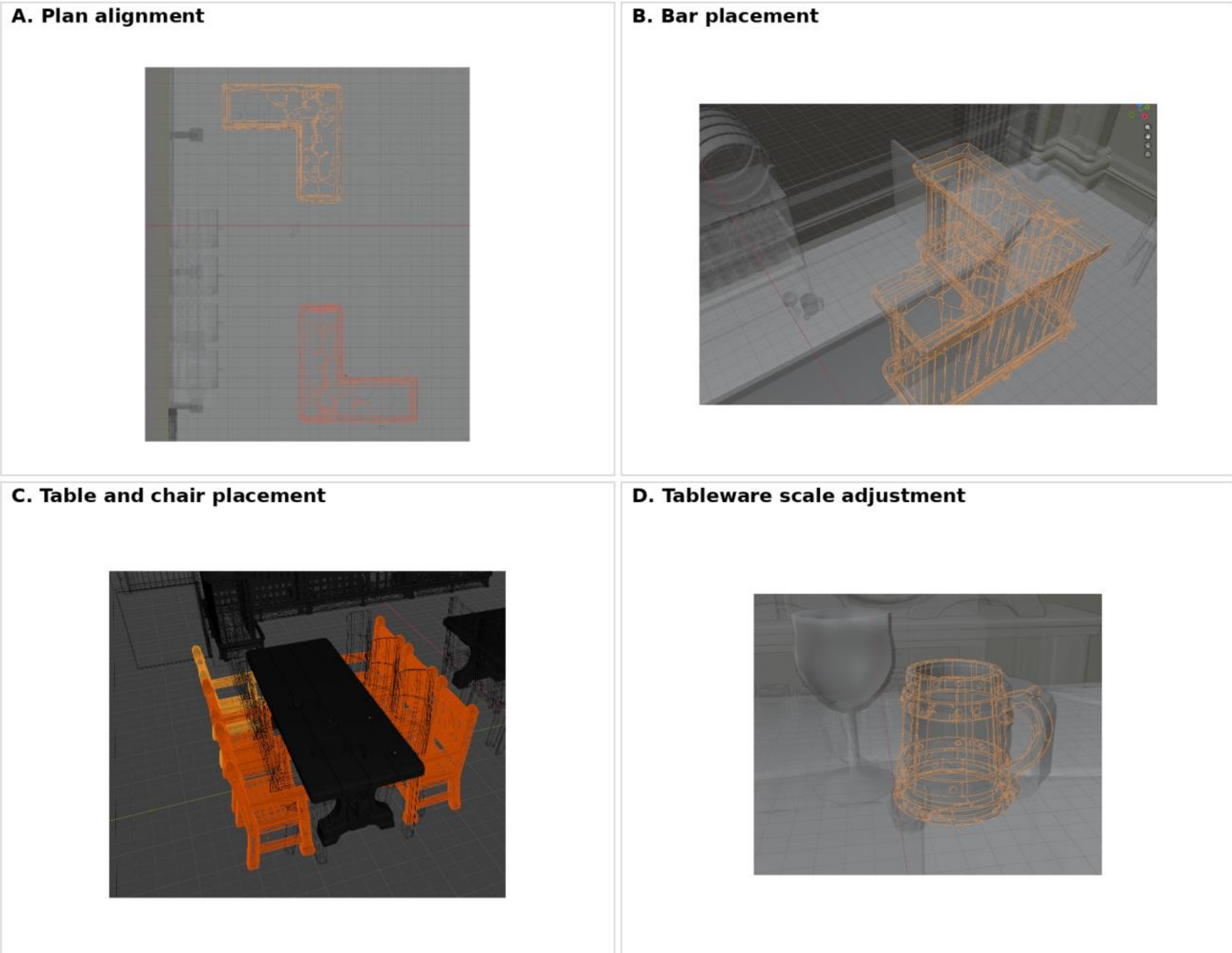


Figure 5. Placement and alignment labor during AI-assisted scene assembly

Figure 6. Final-room comparison between the human-authored and AI-assisted tavern scenes

Taken together, the results support a three-tier viability model. First, AI-generated 3D assets are highly viable for ideation, moodboarding, rapid object exploration, and visual prototyping. In these contexts, prompt speed, variant generation, and surface detail are substantial advantages. Second, AI-generated assets are conditionally viable for static background props when their triangle counts, UV maps, materials, and licensing conditions can be accepted or repaired within a project budget. A generated mug, plate, shelf, or decorative trophy could plausibly be used in a prototype or non-performance-critical scene if optimized or if hardware budgets allow it. Third, AI-generated assets are not yet reliably viable as drop-in replacements for human-authored production assets in performance-sensitive, editable, or interaction-heavy game environments. This does not mean that the tools are failures. It means their most defensible role is as part of a human-guided pipeline in which generation is followed by selection, technical inspection, cleanup, retopology, baking, UV reconstruction, material revision, engine testing, and documentation. The core finding is not simply that AI is faster. The core finding is that speed without asset-readiness can conceal technical debt.

5. DISCUSSION

The study's findings clarify why the phrase 'AI asset creation' can be misleading when used without pipeline specificity. If asset creation means generating a visually plausible object in response to a prompt, then Meshy and Hunyuan performed impressively across much of the tavern asset set. If asset creation means producing an optimized, editable, art-directable, game-ready model, then the same outputs require substantial caution. This distinction echoes Alharthi's (2025) observation that generative AI accelerates ideation and prototyping while also raising concerns about originality, dependency, and human-authored content. The tavern case adds a technical layer to that argument. The difficulty is not only whether AI-generated content is original or aesthetically coherent; it is also whether generated content is structurally responsible. Dense triangulated meshes and fragmented UVs are forms of invisible risk. They may not be obvious in a screenshot, but they become consequential when a scene expands, when optimization targets change, when an asset needs animation, or when a material needs revision. The useful question is therefore not whether AI can create assets, but where in the asset pipeline AI should be allowed to create and where human inspection must intervene.

The concept of technical debt provides a productive way to interpret the time savings observed in the study. Technical debt describes work postponed or hidden by a faster initial decision, later returning as cleanup, maintenance, instability, or integration cost. In the tavern project, AI-assisted generation reduced first-pass production time by nearly two-thirds, but it also introduced debt in mesh density, topology, UV editability, object scale, unwanted features, and optimization planning. The decimation failure is especially illustrative. A dense boar trophy that looks attractive as a generated model can become less useful if automatic simplification creates holes and corrupts texture alignment. The technical debt does not negate the value of the generated model. The high-resolution asset might serve as a concept object, a sculpting reference, a source for baking, or a visual target for a low-poly reconstruction. However, it does mean the asset should not be counted as production-ready merely because it exists. A realistic production workflow would require an artist to retopologize the trophy, reconstruct UVs, bake high-frequency detail into maps, validate materials, and test the result in the engine. That process may still be faster than modeling the object completely from scratch in some cases, but it is a different workflow from direct replacement.

The findings also suggest that text-to-3D tools shift the role of the artist rather than eliminating it. In a traditional workflow, the artist's labor is concentrated in modeling, proportion, topology, UV decisions, material creation, and scene composition. In the AI-assisted workflow, some of the first-pass modeling labor is delegated to the generator, but the artist becomes a prompt designer, curator, technical inspector, optimizer, and integrator. This role shift can benefit students and independent developers because it offers quick access to forms they might not otherwise be able to produce. It can also create risk if users lack the technical literacy to recognize structural problems. A student may see a polished chair and assume it is game-ready, while a technical artist sees excessive triangles, unreadable UVs, inconsistent scale, and difficult collision boundaries. Therefore, game-design education should not teach AI tools as shortcuts around foundational 3D literacy. It should teach them as objects of critique within the asset pipeline. Students should learn to generate assets, but they should also learn to audit them by checking triangle count, topology, material slots, UV maps, texture resolution, file formats, scale, pivot placement, collision setup, and engine performance.

The case study has particular implications for independent game development. Indie teams often face constraints in time, budget, staffing, and specialized expertise. AI-generated assets may appear to solve these constraints by allowing a small team to produce a large asset library quickly. In early prototyping, this advantage is real. A designer can explore the mood of a tavern, test prop density, sketch spatial arrangements, and communicate an art direction before investing in final modeling. However, the same team may later discover that generated assets increase build size, reduce frame rate, complicate texture revisions, or create inconsistent style across a larger scene. The lesson is not to reject AI assets, but to adopt a staged use policy. AI assets can enter the project as references, placeholders, or candidates. They should then pass through a gate that evaluates whether they will be kept as-is, optimized, rebuilt, replaced, or used only as concept material. Such a gate would prevent teams from confusing rapid generation with production completion. It would also help preserve artistic coherence because generated variants can otherwise introduce small stylistic deviations that accumulate across a game world.

Platform governance strengthens the need for documentation. Steam's generative AI content survey distinguishes pre-generated AI content from live-generated AI content and asks developers to describe use of AI services in development or in the product (Valve, n.d.). For projects using AI-generated 3D assets, this means that prompt logs, tool names, model versions, export settings, licensing terms, and asset provenance may become part of a responsible production record. Documentation also supports internal quality control. A team that records prompts, outputs, selected variants, polygon counts, UV status, and correction work can later decide whether a tool is actually saving time. Without such documentation, AI adoption may be evaluated through anecdotal impressions rather than workflow evidence. The tavern project demonstrates the value of a production diary because it reveals both speed and hidden labor. It records that Meshy generated acceptable assets but created download constraints and weak UV maps; it records that

Hunyuan produced attractive models quickly but generated far more triangles than the human-authored scene; and it records that placement labor remained substantial. These observations are precisely the kind of evidence studios need when deciding where AI belongs in their pipelines.
The results, therefore, propose a human-in-the-loop asset pipeline **(Table 4)**. The first stage is ideation, where AI tools generate variants, mood objects, silhouettes, and exploratory forms. The second stage is curation, where artists evaluate prompt adherence, style fit, and object usefulness. The third stage is technical audit, where assets are inspected for triangle count, topology type, UV coherence, material organization, scale, pivot, texture resolution, and export compatibility. The fourth stage is optimization, where artists retopologize, decimate responsibly, bake high-resolution detail, reconstruct UV maps, reduce textures, and prepare LODs. The fifth stage is engine validation, where assets are imported into Unity, Unreal Engine, Godot, or another target engine and tested for frame rate, draw calls, material errors, collision, lighting response, and memory use. The sixth stage is documentation, where licensing, disclosure, prompts, settings, and revisions are recorded. This pipeline does not position AI as a replacement for human artists. It positions AI as a generator of candidate forms whose value depends on the human and technical systems that evaluate them.

Table 4. Human-in-the-Loop Pipeline for AI-Generated 3D Game Assets

Stage	Purpose	Key checks
Ideation	Generate variants, silhouettes, props, and mood objects	Prompt clarity, thematic fit, range of options
Curation	Select assets that fit style and functional requirements	Prompt adherence, unwanted features, art-direction alignment
Technical audit	Determine whether the asset is structurally usable	Triangle count, topology, UV maps, materials, scale, pivot, export format
Optimization	Convert promising assets into efficient game assets	Retopology, UV reconstruction, baking, LODs, texture reduction
Engine validation	Test real-time behavior in target environment	Frame rate, draw calls, memory, import errors, collision, lighting
Documentation	Maintain provenance and disclosure records	Tool name, version, prompts, license, corrections, AI content disclosure

5.1 Implications for Game-Design Education and Production

The findings have direct implications for game-design education because they show that generative AI should be introduced as a technical literacy problem, not merely as a creative convenience. Students often encounter AI tools through impressive preview images, fast browser interfaces, or platform claims about production readiness, which can lead them to equate visual polish with asset completion. A curriculum built around the present case would ask students to generate a model, inspect it in a 3D package, identify polygon count, examine UV layout, evaluate material slots, attempt decimation, and import the model into a game engine. Such an assignment would make the invisible structure of generated content visible. It would also prevent the false opposition between traditional craft and AI use. The goal would not be to ban tools or celebrate them uncritically; it would be to teach students how to distinguish ideation value from production value. In this sense, AI becomes a diagnostic object. It provides students with a concrete artifact that can be interrogated through the vocabulary of topology, scale, shader behavior, file format, collision, and optimization. This approach preserves foundational 3D modeling knowledge while also preparing students for an industry in which generative tools will likely remain part of the production ecology.
For instructors, the tavern case also suggests a useful scaffold for assessment. Instead of grading only the visual appeal of a generated scene, educators can require an asset-readiness report that accompanies every AI-generated object. Such a report might include the prompt, number of generations, selected output, reason for selection, triangle count, texture resolution, UV-map screenshot, topology screenshot, material list, export format, engine-import result, and correction plan. This reporting structure would make process visible and would discourage students from treating generated output as a black box. It would also help instructors evaluate the student's technical judgment rather than the generator's aesthetic capacity. A student who rejects a visually attractive but unusable asset may demonstrate stronger design competence than a student who accepts the most ornate output. Assessment can therefore shift from product worship to pipeline reasoning. The article's evidence supports this pedagogical shift because the Hunyuan room looked visually strong at a glance but produced a triangle burden far exceeding the human-authored room. A classroom that teaches students to see that difference will produce designers who can use AI without surrendering responsibility for the consequences of its output.
For professional studios, the most immediate implication is the need for gatekeeping procedures at the moment AI-generated assets enter a production repository. Studios already use naming conventions, source-control discipline, review boards, style guides, and

technical-art standards to prevent pipelines from accumulating unusable material. AI asset adoption should be governed by similar protocols. A generated asset should not move from concept folder to production folder without an explicit pass through technical review. That review should include whether the asset is placeholder-only, prototype-safe, final-use eligible after optimization, or rejected. Each status category should carry different permissions. Placeholder assets may be used for level blocking and internal demonstrations, but they should not ship. Prototype-safe assets may support playtesting but require planned replacement or optimization. Final-use assets should meet documented budgets and pass engine validation. Rejected assets should be archived only if they provide useful concept reference. Such a system keeps AI output from silently contaminating production builds with inefficient geometry or legally ambiguous material. It also reduces interpersonal conflict because adoption decisions become procedural rather than ideological.

The case further indicates that technical artists may become more, not less, important in AI-assisted pipelines. Popular discourse often imagines generative AI as a way to bypass specialized production labor, yet the tavern project shows that generated assets can increase demand for precisely the skills technical artists possess. Retopology, UV reconstruction, baking, material cleanup, LOD generation, collision design, mesh repair, and engine profiling become essential when a high-detail generated model must become a playable object. A studio that adopts AI without strengthening technical-art review may save time during previsualization but lose time during optimization. Conversely, a studio with strong technical-art support can treat AI models as high-resolution concept sources or sculpting references, then convert promising forms into efficient low-poly assets. This is a more realistic model of productivity than direct replacement. It also clarifies why AI adoption may feel threatening to some artists while simultaneously creating new forms of skilled labor. The labor is not erased; it is redistributed toward judgment, repair, and translation between generated form and production constraint.

Independent developers need a slightly different policy because they often lack separate art, technical art, and engineering departments. For small teams, the danger is not only that AI assets may be technically inefficient, but that inefficiency may remain hidden until late in development. A solo developer can build a prototype quickly with generated props, then discover performance problems only after the scene becomes dense or after target hardware changes. The solution is to build lightweight auditing into the earliest stages. Even a small team can maintain a spreadsheet of generated assets, prompts, triangle counts, texture sizes, export formats, and intended use categories. They can also set simple thresholds: any prop above a particular triangle count must be retopologized; any unreadable UV map must be accepted only for prototypes; any asset with unclear licensing must remain out of commercial builds; any imported model that produces material errors must be flagged before duplication. These small practices transform AI from a risky shortcut into a manageable prototyping resource. They also allow indie developers to make strategic decisions about when to keep, rebuild, or replace generated assets.

The findings also have implications for how tool vendors communicate production readiness. Meshy and Hunyuan demonstrate impressive progress in text-to-3D generation, yet users need more transparent information about what a generated model is structurally prepared to do. A useful generator should expose triangle count, vertex count, UV layout, texture resolution, material channels, object hierarchy, pivot location, scale metadata, and remeshing settings before download. It should also explain the limits of its own optimization claims. If a remesh function can produce quads or reduce a mesh to a target budget, users need to know whether UVs and textures will remain stable, whether baking is performed, whether normals are preserved, and whether the result is appropriate for animation or only for static props. Tool interfaces that foreground these technical details would support responsible adoption. They would also reduce the gap between marketing language and production experience. In this respect, the tavern case should not be read only as a critique of existing generators. It also identifies the kinds of information that future tools should provide if they want to be evaluated as serious components of game-development pipelines.

The study also suggests that research on AI in game development should incorporate more asset-level benchmarks. Many studies examine designer perception, creativity, ideation, narrative generation, or educational use, and those perspectives remain important. However, 3D game production includes measurable technical constraints that can be overlooked when research focuses on prompts and outputs at a high level. A robust benchmark would compare multiple generators across the same asset categories, prompts, art styles, and target engines. It would record not only user satisfaction but also mesh statistics, UV coherence, texture-editing time, retopology time, engine import success, frame rate, draw calls, material errors, and expert technical ratings. Such benchmarks would help developers decide when AI tools are useful and when they introduce more cleanup than they save. They would also help tool designers identify specific failure modes. For example, a generator might perform well on isolated ornamental props but poorly on modular architecture; another might produce beautiful textures but unusable UVs; another might generate efficient geometry but weak prompt adherence. The field needs this level of specificity because production usefulness is always conditional.

Finally, the implications here extend to professional ethics and authorship. AI-generated 3D assets sit at the intersection of creative labor, platform disclosure, intellectual-property uncertainty, and user trust. Developers who use AI-generated assets should therefore maintain prompt records and document tool involvement even when platform rules do not force public disclosure. This practice protects teams by preserving provenance and supporting later review. It also respects the labor ecology of game development by making human and machine contributions traceable. The tavern project demonstrates that human authorship remains embedded throughout the AI-assisted workflow: prompts were written, variants were chosen, assets were scaled, objects were positioned, layouts were adjusted, and technical defects were interpreted by the human designer. Recognizing that labor is important because it resists the simplistic claim that the AI produced the scene alone. At the same time, recognizing AI involvement is also important because the generated geometry and textures were not created through conventional manual modeling. A mature production culture should be able to document both facts at once.

6. CONCLUSION

The fantasy tavern comparison demonstrates that current text-to-3D tools can substantially accelerate early asset production, but it also shows why speed should not be confused with production readiness. Hunyuan 3D generated the asset set in 79 minutes, and the AI-assisted room was assembled in 238 total minutes, compared with 716.06 minutes for the human-authored Blender scene. This difference makes generative AI highly valuable for ideation, rapid prototyping, exploratory scene blocking, and first-pass visualization, especially when teams need fast concept variation across familiar prop categories such as medieval chairs, tavern tables, pewter plates, mugs, fireplaces, barrels, doors, shelves, and wall lanterns. Yet the AI-assisted room also revealed the central technical problem of prompt-based asset production: the generated outputs often looked visually compelling before they were technically ready. The AI room contained far more triangles, fragmented UV maps, dense triangulated meshes, prompt inconsistencies, scale and clipping problems, and failed decimation behavior, while the human-authored workflow took longer but produced geometry and materials that were more legible, editable, and responsive to intentional design decisions. The article's central conclusion is therefore tiered: AI-generated 3D assets are highly viable for ideation, conditionally viable for static background props after technical review and optimization, and not yet reliable as drop-in replacements for human-authored production assets in performance-sensitive game environments.

The contribution of this study lies in its shift from aesthetic comparison to asset-readiness evaluation. Generative AI should not be judged only by whether it can create a medieval-looking chair, table, mug, or fireplace; it should be judged by whether the resulting object can be revised, optimized, textured, imported, disclosed, and maintained within a real game pipeline. The tavern case shows that AI generation relocates labor rather than eliminating it. It moves some work away from first-pass modeling and into prompt iteration, variant selection, cleanup, retopology, UV reconstruction, material correction, scale adjustment, collision setup, engine validation, and documentation. This relocation may still be valuable, particularly for early development, classroom experimentation, small teams, and rapid prototyping contexts, but it requires technical literacy and workflow discipline. A production team cannot simply measure the time between prompt submission and generated preview; it must also measure integration time and production-readiness time. The most useful reporting model would therefore separate three categories: generation time, which measures tool speed; integration time, which measures how long the asset takes to become part of a coherent scene; and production-readiness time, which measures the additional work required before the asset can plausibly ship. A generator that is slower at first-pass output but produces cleaner topology, clearer UVs, and more predictable engine behavior may ultimately be more valuable than a faster system whose assets require extensive reconstruction.

A publishable extension of this study should, therefore, add three layers of original research that move beyond the Blender comparison. First, the human-authored tavern and AI-assisted tavern should be imported into the same game engine under identical conditions, using matched lighting, camera angles, render settings, collision conventions, and target hardware. Such an engine-readiness test should record import errors, material assignment problems, draw calls, frame rate, memory use, file size, triangle and vertex counts, collision-generation time, LOD needs, and manual correction time. This would determine whether the dramatic triangle-count difference translates into measurable runtime consequences. Second, the study should include a blind visual-quality evaluation in which players, game-design students, and technical artists rate matched screenshots or turntable videos for visual appeal, stylistic coherence, fantasy-theme fit, perceived professionalism, perceived readiness for gameplay, and suspected AI use. This distinction matters because an asset can look finished to a general viewer while remaining unusable to a technical artist. Third, an expert technical review should invite 3D artists, environment artists, or game-development instructors to inspect representative assets and rate topology, UV layout, texture editability, mesh density, optimization burden, and likely production use. Combining engine metrics, blind perception data, and expert technical review would strengthen the study by comparing lay perception, artist perception, and measurable production evidence.

Several limitations should frame the interpretation of these findings. The project was produced by a single artist, so the human-production time reflects one individual's skill level, decisions, learning process, and familiarity with Blender. A professional environment artist, technical artist, or novice student might produce different times and different quality outcomes. The scene was also limited to one genre and environment type, a stylized fantasy tavern; results could differ for hard-surface science-fiction props, natural environments, realistic architectural interiors, modular kits, weapons, vehicles, characters, or animated assets. The AI tools tested, Meshy 6 and Hunyuan 3D, were evaluated as accessed during the project period, and generative platforms change rapidly in versioning, pricing, download access, remesh functions, texture quality, and output control. The study also did not complete full engine validation, retopology, LOD generation, runtime benchmarking, or independent participant evaluation, so it identifies strong technical-readiness concerns but does not yet provide final frame-rate, memory-performance, or user-perception data. These limitations do not diminish the value of the case; rather, they clarify the next steps needed to convert a practice-led production benchmark into a multi-method game-development study.

Therefore, the best use of AI in 3D game asset creation is not autonomous replacement but human-in-the-loop production. Generative tools can supply possibilities, accelerate ideation, and produce rapid variations, but skilled artists and technical designers must still determine whether those possibilities can become optimized, editable, ethically documented, and playable assets. The tavern case makes this distinction visible. AI tools can make a room appear quickly, but human expertise determines whether that room can function as part of a game. The future of AI-assisted game-asset production will depend less on whether models look impressive in isolation and more on whether they can survive the full pipeline from prompt to prop, from prop to scene, and from scene to playable environment.

REFERENCES

1. Alharthi, S. A. (2025). Generative AI in game design: Enhancing creativity or constraining innovation? *Journal of Intelligence, 13*(6), 60. https://doi.org/10.3390/jintelligence13060060
2. Begemann, A., & Hutson, J. (2024). Empirical insights into AI-assisted game development: A case study on the integration of generative AI tools in creative pipelines. *Metaverse, 5*(2), 2568. https://doi.org/10.54517/m.v5i2.2568
3. Begemann, A., & Hutson, J. (2025). Navigating copyright in AI-enhanced game design: Legal challenges in multimodal and dynamic content creation. *Journal of Information Economics, 3*(1), Article 42. https://doi.org/10.58567/jie03010001
4. Chen, Z., Tang, J., Dong, Y., Cao, Z., Hong, F., Lan, Y., Wang, T., Xie, H., Tong, W., Saito, S., Pan, L., Lin, D., & Liu, Z. (2024). 3DTopia-XL: Scaling high-quality 3D asset generation via primitive diffusion. *arXiv*. https://arxiv.org/abs/2409.12957
5. Costa, G. P. (2026). The Role of Game Engines in the Democratization of Digital Game Development. *Journal International Review of Research Studies*, *1*(02), 1-17.
6. French, F., Levi, D., Maczo, C., Simonaityte, A., Triantafyllidis, S., & Varda, G. (2023). Creative use of OpenAI in education: Case studies from game development. *Multimodal Technologies and Interaction, 7*(8), 81. https://doi.org/10.3390/mti7080081
7. Fukaya, K., Daylamani-Zad, D., & Agius, H. (2026). Heuristics for AI-Driven Graphical Asset Generation Tools in Game Design and Development Pipelines: A User-Centered Approach. *International Journal of Human–Computer Interaction*, 1-22.
8. Game Developers Conference. (2026). GDC 2026 State of the Game Industry reveals impact of layoffs, generative AI, and more. https://gdconf.com/article/gdc-2026-state-of-the-game-industry-reveals-impact-of-layoffs-generative-ai-and-more/
9. Holzner, N., Maier, S., & Feuerriegel, S. (2025). Generative AI and creativity: A systematic literature review and meta-analysis. *arXiv*. https://arxiv.org/abs/2505.17241
10. Kuusela, V. (2022). 3D modeling pipeline for games [Bachelor's thesis, Turku University of Applied Sciences]. *Theseus*. https://urn.fi/URN:NBN:fi:amk-2022052311120
11. Lai, Z., Zhao, Y., Liu, H., Zhao, Z., Lin, Q., Shi, H., Yang, X., Yang, M., Yang, S., Feng, Y., Zhang, S., Huang, X., Luo, D., Yang, F., Yang, F., Wang, L., Liu, S., Tang, Y., Cai, Y., He, Z., et al. (2025). Hunyuan3D 2.5: Towards high-fidelity 3D assets generation with ultimate details. *arXiv*. https://arxiv.org/abs/2506.16504
12. Lin, Z., Ehsan, U., Agarwal, R., Dani, S., Vashishth, V., & Riedl, M. (2023). Beyond prompts: Exploring the design space of mixed-initiative co-creativity systems. *arXiv*. https://arxiv.org/abs/2305.07465
13. Meshy LLC. (n.d.). Meshy AI: The easiest way to create 3D models. https://www.meshy.ai/
14. Mikalonyte, E. S., & Kneer, M. (2022). Can artificial intelligence make art? Folk intuitions as to whether AI-driven robots can be viewed as artists and produce art. *ACM Transactions on Human-Robot Interaction, 11*(4), 1-19. https://doi.org/10.1145/3530875
15. Moon, J., Lee, U., Koh, J., Jeong, Y., Lee, Y., Byun, G., & Lim, J. (2024). Generative artificial intelligence in educational game design: Nuanced challenges, design implications, and future research. *Technology, Knowledge and Learning, 30*, 447-459. https://doi.org/10.1007/s10758-024-09756-z
16. Nagpal, P., Chugh, A., & Rohilla, O. (2026). G-GenAI: An AI-Driven Framework for Co-Creative Game Development. In *Examining AI Innovations in Gaming and Entertainment* (pp. 155-178). IGI Global Scientific Publishing.
17. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.-Y., Skorokhodov, I., Wonka, P., Tulyakov, S., & Ghanem, B. (2023). Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. *arXiv.* https://arxiv.org/abs/2306.17843
18. Ramos, L. A. G. T. (2017). *Production of 3D Animated Short Films in Unity 5: Can Game Engines Replace the Traditional Methods?* (Master's thesis, Universidade Catolica Portuguesa (Portugal)).
19. Savery, R. (2026). Video Game Development. In *Handbook of AI-based Media Disruption: Emerging Technologies and Multidisciplinary Perspectives Across Sectors* (pp. 137-155). Cham: Springer Nature Switzerland.

20. Ternar, D. A., Denisova, A., Cunha, J. M., Kultima, A., & Guckelsberger, C. (2026, April). Generative AI in Game Development: A Qualitative Research Synthesis. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (pp. 1-24).
21. Valve. (n.d.). Content survey. *Steamworks Documentation*. https://partner.steamgames.com/doc/gettingstarted/contentsurvey
22. Wadinambiarachchi, S., Kelly, R. M., Pareek, S., Zhou, Q., & Velloso, E. (2024). The effects of generative AI on design fixation and divergent thinking. arXiv. https://arxiv.org/abs/2403.11164
23. Werning, S. (2024). Generative AI and the technological imaginary of game design. In *Creative tools and the softwarization of cultural production* (pp. 67-90). Cham: Springer Nature Switzerland.
24. Yannakakis, G. N., & Togelius, J. (2018). Artificial intelligence and games. Springer. https://doi.org/10.1007/978-3-319-63519-4
25. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., Shi, H., Liu, S., Wu, J., Lian, Y., Yang, F., Tang, R., He, Z., Wang, X., Liu, J., Zuo, X., et al. (2025). Hunyuan3D 2.0: Scaling diffusion models for high resolution textured 3D assets generation. *arXiv*. https://arxiv.org/abs/2501.12202