

AI Game Melding: Agent-Assisted Reconstruction, Player Co-Creation, and the Future of Programmable Play

Andrew Allen Smith¹ & James Hutson^{2*}

¹Communications, College of Arts and Humanities, Lindenwood University

^{2*}Human Cultures, College of Arts and Humanities, Lindenwood University

Received: 05 / 09 / 2026

Accepted: 09 / 10 / 2026

Published: 21 / 10 / 2026

Abstract: Artificial intelligence increasingly mediates game production and player participation, but claims about game merging frequently conflate different technical achievements. This article examines AI game melding as the convergence of agent-assisted reconstruction, executable interoperability, and creative recombination. An integrative narrative review connects procedural generation, computational blending, modding, co-creation, and decompilation research with public documentation available through October 15, 2026. Cases include meme-based and Wii Sports crossovers in World at War Zombies, a Call of Duty and Minecraft hybrid, runtime bridging, and agent-assisted reconstruction. The analysis separates creator demonstrations and reported acceleration from independently validated performance. It proposes a five-layer composability stack comprising representation, semantics, runtime, verification, and permission. Reduced implementation friction may relocate production bottlenecks, making behavioral validation, design coherence, maintenance, and rights governance increasingly consequential. Four conditional scenarios are considered: expanded bespoke play, abundance accompanied by discoverability problems, stronger enclosure, and governed experimentation. The literature gap concerns limited integrated evidence linking substantial game reconstruction and interoperability to community, economic, and governance outcomes, rather than an absence of earlier blending research. Proposed studies address acceleration, player experience, attribution, preservation, and platform power. Game melding may redistribute creative authority, but its consequences depend on institutions and verification practices surrounding programmable play rather than technical capability alone.

Keywords: generative artificial intelligence; game melding; modding; decompilation; human-AI co-creation; platform governance.

Cite this article: Smith, A. A. & Hutson, J. (2026). AI Game Melding: Agent-Assisted Reconstruction, Player Co-Creation, and the Future of Programmable Play. *MRS Journal of Multidisciplinary Research and Studies*, 3(10), 26-41.

Introduction

Artificial intelligence is changing gaming at several distinct levels: how developers produce games, how virtual characters respond, how interactive environments emerge, and how players modify existing software. Developments during 2026 illustrate these differences more clearly than predictions about an undifferentiated, fully automated game industry. On January 29, Google introduced Project Genie, an experimental application for creating, exploring, and remixing interactive worlds from prompts and images; its launch documentation also described constraints on duration, control, and consistency (Rivas et al., 2026). On February 4, Roblox announced an open beta for Cube-powered “4D” generation, extending generation toward interactive objects rather than merely static three-dimensional assets (Roblox, 2026). NVIDIA subsequently described the June 17–30 public beta of PUBG Ally, an AI teammate for the battle royale shooter PlayerUnknown’s Battlegrounds (PUBG), available through a limited testing arrangement rather than a universal replacement for human collaborators (Burnes, 2026). These examples concern world simulation, creator tooling, and in-game companionship, respectively. None independently establishes that commercially complete games can be generated without sustained human design. Their importance lies instead in the expansion of tasks that software can mediate between an intention and a playable result.

PUBG’s presence in this account is particularly significant because its development emerged from an earlier culture of modification. Brendan Greene’s route to the standalone game began with battle royale modding in the Arma ecosystem (Francis, 2017). The example connects an established form of player-led innovation with a later commercial platform for AI experimentation. Fortnite’s trajectory likewise illustrates how a recognizable play structure can be adapted rather than reproduced without alteration. Epic introduced its separate Battle Royale mode in 2017, explicitly acknowledging PUBG as an inspiration while retaining Fortnite’s building mechanics (Mustard, 2017). PUBG Corporation withdrew its lawsuit against Epic in 2018; that withdrawal was not a judicial ruling on the merits (Kim, 2018). The sequence places creative borrowing and disputes over control within the same history. AI game melding therefore enters an industry already shaped by the movement of mechanics between communities and commercial productions.

An earlier mainstream experiment also exposed the tension between conversational freedom and developer control. On May 16, 2025, Fortnite introduced a recruitable Darth Vader character with whom players could converse by voice (StarWars.com Team, 2025). The licensed crossover connected a familiar fictional

identity to generated responses rather than only to prerecorded dialogue. Players rapidly attempted to elicit prohibited language, and WIRED documented profanity and other problematic responses (Farokhmanesh, 2025). Epic told the publication that it deployed a corrective update within 30 minutes of one reported incident. These interactions tested the character's guardrails, not the recovery or rewriting of Fortnite's source code. Their relevance to melding lies in the contest over what users can make an authorized system express. The example shows why expanding participation also requires distinguishing productive reinterpretation from attempts to defeat the boundaries of a designed experience.

That expansion changes the practical meaning of authorship without eliminating the differences between a compelling demonstration and a durable game. A generated landscape may support exploration while lacking a coherent progression system, dependable persistence, or multiplayer synchronization. An articulate companion may remain constrained by the actions its host game permits. A generated object may require collision testing, animation adjustment, and integration into an existing economy. These distinctions suggest that AI capability should be assessed at the level of a production task and its surrounding dependencies, rather than inferred from a visually impressive output. Industry sentiment also resists a simple narrative of inevitable enthusiasm: the 2026 Game Developers Conference survey reported that 52% of respondents regarded generative AI as having a negative industry impact, compared with 30% in the preceding survey (Game Developers Conference [GDC], 2026). These responses describe the surveyed population, not a causal estimate of employment effects or a representative verdict from every game developer. They nevertheless establish that growing technical capability coexists with unresolved concerns about creative work and its institutional organization.

A more disruptive development occurs when AI assistance moves from creating assets for a game to reconstructing, connecting, or transforming the executable systems of several games. P. E. Cooper's October 2026 posts present Call of Duty: World at War Zombies crossovers with Fortnite and Wii Sports, making this emerging practice culturally legible through familiar combinations (Cooper, 2026b, 2026c). SkyCraft describes a different arrangement: a bridge between running instances of Skyrim and Minecraft, not the replacement of both by a newly decompiled engine (chasmlol, 2026b). Such projects motivate the term AI game melding: AI-assisted recombination that changes relationships among game content, operational rules, software implementations, and player authorship. The term does not designate one algorithm or imply that every project uses the same model. It identifies a sociotechnical problem connecting game design, reverse engineering, participatory production, and platform governance. This article asks what becomes possible when executable relationships become easier to reconstruct, and who acquires practical control over those possibilities. Its argument concerns the distribution of composability, verification, and permission, not simply the automation of content production.

The resulting future cannot be reduced to either universal bespoke games or the disappearance of modding behind streamed services. Those outcomes are useful boundary scenarios, but each suppresses important technical and institutional variation. A publisher might license controlled combinations, maintain an accessible client with protected online systems, or support

preservation while restricting commercial redistribution. A player might prefer a personally tailored experience for one purpose and a stable, collectively authored game for another. The article therefore extends earlier work on adaptive worlds and collaborative game development by examining what happens when co-creation crosses the boundary between distinct executable works (Ratican & Hutson, 2024a, 2024b). It first reconstructs the relevant literature, then develops a technical taxonomy and critically examines contemporary cases. The subsequent discussion evaluates conditional industry futures and proposes a research agenda that could discriminate among them. The objective is neither to celebrate every remix nor to defend enclosure categorically, but to identify the conditions under which expanded creative participation can remain technically credible, culturally meaningful, and institutionally sustainable.

Literature Review

From Designed Behavior to Learned Content

The history of AI in games is not a succession in which machine learning simply replaces primitive scripting. Search, planning, rule systems, behavioral organization, and procedural generation address different design problems and can coexist within the same game. Togelius et al. (2011) organized search-based procedural content generation around representations, evaluation, and search processes, emphasizing that generation depends on what a system can describe and reward. This perspective remains valuable because a generator does not independently determine the purpose of an experience. Its output reflects constraints and evaluative assumptions, whether those assumptions are explicit rules or learned preferences. Accordingly, a large environment does not necessarily constitute an adaptive narrative, and procedural variation does not automatically imply learning from an individual player. For game melding, the enduring lesson is that imported elements must satisfy an evaluative structure larger than their own local operation. Combining two successful mechanics may produce a poor game when their incentives, temporal rhythms, or resource assumptions conflict. The historical continuity lies in that design problem, even when the means of searching its possibilities become more powerful.

Machine learning added another route to procedural generation by learning relationships from existing game content rather than relying exclusively on hand-specified generators. Summerville et al. (2018) characterized this approach as procedural content generation via machine learning, encompassing the production and transformation of content through trained models. Their framework situates generative systems within a longer trajectory of representation learning rather than treating contemporary language models as an isolated beginning. For the present analysis, the important shift is that examples can become a resource for constructing a space of possible designs. That space, however, depends on the features represented and the content available for training. It may encode visual patterns effectively while representing causal rules poorly. A generator can therefore preserve the appearance of a genre without preserving the conditions that make its challenges intelligible. This distinction helps explain why increasingly convincing assets do not, by themselves, demonstrate increasingly coherent games. It also establishes a conceptual bridge to melding: the central question becomes which relationships between source works a system can represent, recombine, and validate.

Modding, Participatory Labor, and Existing Boundaries

Modding established player participation in game production long before contemporary generative AI. Scacchi (2010) examined modding as an ecosystem of modifications, communities, practices, and technical arrangements rather than an activity reducible to editing files. Nieborg and van der Graaf (2008) likewise situated nonmarket game production within industrial relationships, challenging the assumption that participation necessarily occurs outside commercial organization. Together, these accounts explain why the availability of a tool does not fully determine creative autonomy. Participants depend on documentation, interfaces, distribution channels, social legitimacy, and continuing access to the underlying game. Those dependencies can enable experimentation while concentrating strategic power elsewhere. In the present context, AI changes some costs within this relationship, especially the effort required to interpret unfamiliar code or attempt an integration. It does not automatically transfer ownership of infrastructures or decision rights over distribution. The analytical implication is that democratization must be evaluated through the institutions surrounding production, not inferred from the number of people who can generate a prototype.

Several influential commercial games developed from precisely this participatory setting. Counter-Strike began in 1999 as a tactical first-person shooter modification of Half-Life (Guinness World Records, n.d.). Dota 2's lineage runs through Defense of the Ancients, a Warcraft III custom map whose development relied on the World Editor and community contributions (Mescon, 2009; Schreier, 2010). Greene's battle royale work similarly moved through Arma 2 and Arma 3 before PUBG, with Bohemia Interactive recognizing his Arma 3 Battle Royale project in its modding competition (Bohemia Interactive, 2015; Francis, 2017). These examples show that mods can become design foundations rather than remain supplementary content. They also demonstrate that modification does not always require decompiling an entire game. Official editors, scripting interfaces, and accumulated community knowledge can provide alternative entry points. AI assistance may broaden participation by helping newcomers interpret those tools and construct changes, but it does not erase the expertise required for difficult reconstruction or reliable integration.

Modder motivations further complicate predictions that faster production will either destroy craftsmanship or simply expand an interchangeable supply of content. Sotamaa (2010) documented the plurality of motivations and practices within modding culture, offering an alternative to explanations based solely on economic compensation or technical achievement. This perspective makes room for learning, participation, personal expression, and attachment to a particular game as analytically distinct interests. Applied to AI-assisted work, it suggests that identical tools may have different meanings for different participants. A creator who values understanding an engine may experience extensive automation differently from someone seeking a specific accessibility modification. Neither preference can be inferred from the speed of the resulting build. The appropriate question is what agency a workflow preserves and what forms of participation it makes possible or less attractive. Similarly, the health of a community cannot be measured only by its output volume. A larger number of releases could coexist with declining mentorship, weaker attribution, or stronger newcomer

participation; these possibilities require observation rather than assumption.

Game Blending Before the Current Wave

Automated recombination is also an established research topic, which limits any defensible claim that game merging has never been discussed. Guzdial and Riedl (2018) investigated automated game design through conceptual expansion, using computational representations to move beyond direct reproduction of an existing game. Their work demonstrates that combining and transforming learned design knowledge predates current coding agents. The contribution is important here because it shifts attention from isolated assets toward relationships that organize a playable system. However, computational game design research and contemporary commercial-game reconstruction do not address identical objects. A model can explore a constrained design representation without recovering a proprietary runtime or integrating its external dependencies. Conversely, a runtime bridge can connect existing systems without learning a general representation of game design. Recognizing both continuity and difference prevents two symmetrical errors: overstating the novelty of recombination and understating the practical significance of new executable integrations. The emerging phenomenon requires a more precise unit of comparison than the claim that two games have been combined.

Latent-space research provides a second direct precedent for blending across game domains. Sarkar and Cooper (2022) examined latent combinational game design, demonstrating a computational approach to generating combinations from multiple source domains. Such work treats combinations as locations or relationships within learned representational spaces. It therefore offers a way to reason about controllable variation rather than viewing every hybrid as an unstructured collage. Yet representational blending does not imply that source binaries, commercial assets, or live network services have become interoperable. That difference matters because each imposes distinct technical and legal dependencies. The present article uses this distinction to separate design-space recombination from executable recombination, while recognizing that future systems could combine both. An agent might first propose a hybrid mechanic and then construct the software needed to realize it. The novelty would lie in that coordinated workflow and its deployment conditions, not in the previously established idea that game characteristics can be computationally blended.

Generative Agents, World Models, and Co-Creation

Research on generative agents extends the discussion from generating objects to organizing behavior and social interaction. Park et al. (2023) developed an architecture combining memory, reflection, and planning to support believable behavior among simulated agents. This work helps explain why language-mediated interaction has attracted interest beyond scripted dialogue. Nevertheless, believable social responses do not establish reliable game progression, balanced resource systems, or coherent relationships among imported mechanics. These remain separate design and verification problems. In a melded game, an agent may describe actions that one host engine permits but another cannot represent. The resulting discrepancy concerns operational authority, not merely conversational quality. Accordingly, generative characters should be treated as components whose intentions must connect to explicit action interfaces and persistent

world states. This analytical separation preserves the value of language-mediated interaction while preventing social fluency from becoming a proxy for the integrity of the surrounding game.

World models introduce a further distinction because they can simulate interactive appearances without reconstructing the original software implementation. GameNGen, first presented in 2024 and published at ICLR in 2025, generated action-conditioned Doom gameplay through a diffusion-based approach (Valevski et al., 2025). Its relevance concerns learned simulation, not recovery of the original source code. A sequence of responsive frames and a conventionally implemented game may look similar while providing different access to state, rules, debugging, and modification. For this article, that distinction is more important than treating one architecture as inherently superior. A designer seeking an exploratory visual experience may tolerate properties that would be unacceptable for competitive play or exact preservation. Conversely, an explicit engine can expose editable systems while demanding substantial engineering work. Game melding could eventually employ either architecture, but their affordances should not be conflated. Claims about the future of modding must therefore specify whether users can manipulate representations, interfaces, executable logic, or only the prompts that influence a remote generator.

Earlier work by Ratican and Hutson (2024a, 2024b) provides the immediate conceptual foundation for examining this redistribution of creative control. Adaptive Worlds foregrounded personalization and the changing relationship between players and generated environments, while Video Game Development 3.0 emphasized AI-supported collaborative production. Together, these publications identified co-creation as a useful alternative to treating AI solely as a labor-saving addition to established pipelines. The present article retains that orientation but disaggregates the technical pathways through which participation occurs. Adaptation during play, generation during development, and reconstruction after release should not be treated as equivalent achievements. Each has different dependencies, evaluative criteria, and consequences for authorial control. Moving from co-creation within one game to recombination across games also introduces conflicts between design systems that were not intended to share authority. The extension proposed here is therefore specific: AI-mediated participation becomes a problem of cross-system coordination and permission. This shifts the analysis from whether players can create more content to whether they can meaningfully reorganize the executable relationships that constitute an experience.

The production and governance dimensions of this extension are already visible in Hutson's collaborative research with Begemann. Their game-development case study examined the integration of generative tools within creative pipelines, providing a situated account rather than a controlled industry-wide productivity estimate (Begemann & Hutson, 2024). Their subsequent copyright analysis emphasized the multimodal and dynamic character of games and the importance of human contribution and provenance (Begemann & Hutson, 2025). These concerns become more complicated when a project combines multiple existing works instead of generating assets for a single production. A useful extension must therefore distinguish the origin of an element from the permission to adapt it, and distinguish technical integration from ownership. It must also consider contributions that remain embedded in tools and

documentation rather than visibly appearing in a final scene. Such contributions can be indispensable even when a new agent performs the most conspicuous integration work. The proposed framework accordingly treats attribution, dependency tracking, and verification as part of the creative pipeline. They are not administrative matters that begin only after the game has been assembled.

Recent human-computer interaction research cautions against equating generative assistance with unconstrained creative benefit. Alharthi (2025) examined the tension between creative support and constraint, while Ternar et al. (2026) synthesized qualitative research on generative AI in game development. The latter integrated ten studies and identified issues involving agency, workflows, and the practical accommodation of tools. These publications support examining situated use rather than inferring outcomes from model demonstrations. An especially relevant 2026 example is SPINE, whose design-pillar interface allows people to evaluate language-model feedback within a mixed-initiative workflow (Geheeb et al., 2026). Its relevance is the explicit organization of human judgment, not proof that the particular interface solves game melding. A hybrid project needs comparable mechanisms for recording why a combination was accepted and which constraints it must preserve. Otherwise, repeated generation may enlarge the design space while making the rationale for a chosen design increasingly difficult to recover. The literature thus directs attention toward the quality of collaboration, not simply the amount of generated material.

Decompilation and the Verification Problem

The historical labor involved in decompilation helps explain why recent demonstrations attract attention within modding communities. Robinson (2021) reported that the Super Mario 64 reconstruction released in 2019 followed a two-year effort. Even this older game required sustained work to recover usable higher-level representations from a compiled program. Such a reported duration is a historical comparison, not a controlled estimate of the speedup delivered by contemporary AI tools. The technical accounts reviewed below explain why reconstruction remains demanding: structural recovery, naming, behavioral fidelity, and integration impose different burdens. Greater system complexity can increase those burdens, although release date alone does not determine difficulty (Tan et al., 2024, 2026). This labor-intensive setting suggests a selection effect worth testing: games with large, committed communities may attract more reconstruction effort than less familiar titles. If AI reduces the expertise or time needed for particular stages, obscure games could become more practical subjects for modification and preservation. That possibility broadens the significance of acceleration beyond faster work on already prominent franchises.

LLM-assisted decompilation adds a distinct technical strand to this history. Tan et al. (2024) introduced LLM4Decompile, evaluating approaches that generate higher-level code from lower-level representations or refine the output of conventional decompilers. Their work distinguishes readable output from output that can execute correctly under an evaluation procedure. The refinement architecture combines an established decompiler with a language model in complementary roles. This arrangement contradicts accounts that erase the contribution of existing reverse-engineering tools. It also clarifies why decompilation cannot generally be understood as retrieving a hidden copy of the developer's original source text. Compilation

can discard names, comments, and structural information, leaving several plausible reconstructions compatible with observed behavior. For game research, this means that reconstruction quality requires explicit evidence about behavior and scope. A benchmark result on functions does not establish full-game fidelity, nor does readable code establish the preservation of every relevant system relationship.

Two-phase decompilation research further illustrates the importance of separating objectives that are often collapsed in public discussion. SK²Decompile distinguishes structural recovery from identifier naming, using an intermediate representation to connect the phases (Tan et al., 2026). The corresponding preprint makes this separation explicit in its architectural account (Tan et al., 2025). The methodological value lies in recognizing that producing meaningful names and preserving program structure are related but nonidentical tasks. A persuasive reconstruction can be easy to read while expressing an incorrect relationship among data fields or control paths. Conversely, an unattractive intermediate representation may preserve useful constraints that should not be discarded prematurely. Applied to game melding, the same principle suggests separating interpretation, adaptation, and validation rather than asking one generative pass to accomplish everything. Such separation makes responsibility and failure easier to localize. It also provides a basis for comparing workflows: the strongest system is not necessarily the one that produces the most polished-looking source immediately, but the one whose transformations remain inspectable and testable.

Evidence published in September 2026 strengthens the case for caution about apparently successful reconstruction. Liu et al. (2026), in a preprint rather than a peer-reviewed game study, reported cases in which decompiled functions passed supplied tests but diverged from their originals under additional behavioral comparison. Their analysis challenges the sufficiency of compilation and fixed test-suite success as universal indicators of preservation. The relevance to games is inferential: a complex interactive system contains many states that a brief demonstration will not visit. A reconstructed mechanic might function during ordinary movement yet fail when saving, loading, changing equipment, or interacting with another imported system. That possibility does not show that any particular public mod contains such defects. It identifies a reason why evidence standards must grow with the strength of the claim. Assertions about faithful reconstruction therefore require broader testing than assertions about a promising prototype. Assertions about commercial readiness require additional evidence concerning usability, maintenance, security, and the experience of actual players.

The Specific Gap

The literature therefore does not support an absolute claim that researchers have ignored AI-mediated game recombination. Computational blending, participatory production, generative co-creation, and learned decompilation each have established bodies of work. What remains insufficiently integrated in the sources examined here is their convergence in agent-assisted projects that reconstruct or connect substantial commercial-game systems. In particular, this review did not identify a comparative, longitudinal study that jointly evaluates whole-game reconstruction, executable interoperability, inherited community labor, rights arrangements, and subsequent industry outcomes. This is a bounded finding about the reviewed literature, not proof that no relevant study exists anywhere. It identifies a research problem whose components are

known but whose interaction demands closer investigation. The distinctive contribution of the present article is to define that intersection, differentiate its technical forms, and specify competing explanations for its consequences. The gap concerns evidence about how executable recombination changes creative control under real production and governance conditions. Addressing it requires connecting the technical question of what runs with the cultural and institutional questions of who can modify, maintain, distribute, and benefit from it.

Methodology

This article uses an integrative narrative review and a comparative analysis of public technical documentation, with a research cutoff of October 15, 2026. The scholarly strand connects procedural generation, computational blending, modding studies, human-AI co-creation, and decompilation research. The contemporary strand examines developer accounts, project repositories, official product announcements, creator posts, contemporaneous reporting, and relevant policy materials. Searches combined terms such as game blending, game merging, AI-assisted modding, Claude decompilation, runtime interoperability, generative game development, and language-model decompilation. Citation following connected recent demonstrations to earlier research and to the prior publications that this article extends. Creator posts establish public framing, while reporting supplies attributed descriptions rather than technical validation. Primary project documentation was preferred for architectural claims; official announcements were used to establish what organizations had publicly introduced or proposed. Preprints were retained where they addressed recent technical developments, but their publication status is stated. This is not a preregistered systematic review, and no exhaustive database coverage or reproducible screening count is claimed.

The comparative analysis distinguishes reported capability, inspectable documentation, and independently validated performance. A repository description can establish what its developer claims to have built and how the project is organized, but it does not by itself confirm every implementation claim. A demonstration can establish that a selected sequence was shown, but it cannot establish long-term stability or general performance. A developer's elapsed-time report is also not a controlled comparison with a human-only baseline. Accordingly, the analysis records the object being transformed, the apparent integration mechanism, the prerequisites, and the limits of the available evidence. It does not execute the projects, audit their complete codebases, or reproduce their benchmarks. The supplied YouTube link served as a discovery prompt rather than an evidentiary substitute for accessible technical sources. Three user-supplied gameplay stills are analyzed as visual documents and reproduced as Figures 1–3 with source credits. They establish visible presentation and interface combinations, not runtime architecture or independent verification of AI-development claims.

The interpretive procedure develops conditional explanations rather than predicting a single industry outcome. It asks how changes in reconstruction cost might interact with verification demands, creative preferences, intellectual-property arrangements, and the distribution of infrastructure. Cases were chosen for architectural contrast and visible creative experimentation, not as a representative sample of modding activity. This selection is appropriate for clarifying concepts but cannot estimate adoption rates or the prevalence of failures. The

proposed scenarios should consequently be understood as analytical possibilities with observable indicators, not probabilistic forecasts. Likewise, the research agenda specifies studies that remain to be conducted rather than presenting proposed measures as collected data. These boundaries are essential because rapidly changing software ecosystems reward dramatic claims more quickly than careful replication. The article's findings are therefore a conceptual synthesis and an evidence-qualified interpretation of contemporary documentation.

Analytical Results: A Framework for Executable Recombination

Distinguishing the Object Being Combined

The first analytical result is that game melding requires a taxonomy of operations rather than a single scale running from manual modding to autonomous creation. Table 1 distinguishes asset transplantation, learned design blending, runtime bridging,

reimplementation, functional decompilation, and binary-matching reconstruction. These categories can overlap within one project, but they answer different questions. Asset transplantation changes what is represented; a runtime bridge changes how existing processes communicate; reconstruction changes which implementation supplies behavior. A world model introduces another possibility by learning an interactive simulation without necessarily exposing a conventional source-level engine. None of these operations guarantees that the result is a new game in the same sense. A cosmetic replacement may preserve nearly all of the host's rules, whereas a novel relationship between movement, resources, and objectives may change the experience substantially. The taxonomy therefore separates the technical operation from the cultural judgment of novelty. It also prevents elapsed times from being compared across projects that have accomplished fundamentally different tasks.

Table 1 Technical Operations Commonly Collapsed Under Game Melding

Operation	Object transformed	What it does not establish
Asset transplantation	Models, textures, audio, or other representations	Imported mechanics or a reconstructed engine.
Learned design blending	A learned representation spanning game domains	Interoperability between original commercial runtimes.
Runtime bridging	Communication between existing running systems	Recovery or replacement of both source implementations.
Reimplementation	A new implementation of selected behavior	Exact recovery of original code or complete behavioral parity.
Functional decompilation	Higher-level code reconstructed from compiled representations	Identity with the original source or correctness on every input.
Binary-matching reconstruction	A reconstruction evaluated against specified compiled outputs	Commercial readiness, distributable asset rights, or successful new game design.
World-model simulation	An action-conditioned learned simulation	A conventional editable engine or source-level reconstruction.

Shared-memory passthrough describes one concrete mechanism within the runtime-bridging category. Bridge code can expose an agreed region of system memory through which separate game processes exchange position, state, and collision information. SkyCraft's design document specifies this arrangement alongside synchronization events and shared GPU textures (chasmlol, 2026b). Skyrim supplies nearby world geometry and actor data, while Minecraft supplies player position, orientation, and other state. Visual layers are shared through a related graphics pathway rather than by assuming that all asset files pass through one memory buffer. Coordinate conversion and an agreed message format allow each program to interpret the other's data. AI may assist in writing these bridges, but the resulting software still needs explicit authority rules, synchronization, and testing. The term therefore identifies a form of interoperability, not proof that two complete source-code repositories have been recovered or that their engines have become one.

A second result concerns the location of authority within a hybrid system. When two engines participate, each may assume that it controls movement, time, object identity, and persistence.

Those assumptions cannot all remain authoritative when their outputs conflict. Consider a hypothetical hybrid in which one engine calculates a character's trajectory while another detects collisions and stores the save state. A momentary visual agreement could conceal a disagreement about where the character is allowed to stand or what state should persist after reloading. The integration problem is therefore not exhausted by transferring values between processes. It requires an explicit allocation of authority and a policy for resolving mismatches. This analysis suggests that a game can become more visually integrated without becoming more semantically coherent. Evaluating melding should consequently include tests of transitions and disagreement, not only tests of the individual systems operating under ideal conditions.

A Five-Layer Composability Stack

The proposed composability stack organizes these dependencies into five layers: representation, semantics, runtime, verification, and permission. Representation concerns formats, assets, identifiers, and descriptions that one component can recognize. Semantics concerns what actions and objects mean, including their consequences within rules and progression.

Runtime concerns execution, synchronization, persistence, and the authority assigned to participating systems. Verification concerns the evidence that the assembled system behaves as intended across relevant conditions. Permission concerns the rights and institutional arrangements that authorize access, modification, distribution, and maintenance. These layers are analytically separable even when a practical workflow interleaves them. Progress at one layer does not establish progress at every other layer. An agent might translate a format successfully while leaving a semantic conflict unresolved, or construct a functioning bridge that cannot lawfully be distributed with the required assets. The stack makes these discontinuities visible and provides a structure for reporting them rather than hiding them behind a single claim of successful merging.

This framework also distinguishes access to a game from control over its future. Possession of a client executable does not mean possession of the original development repository, build system, server implementation, or asset licenses. Conversely, the absence of original source code does not establish that modification is impossible. The practical issue is which layers remain accessible and under what conditions. An official interface may provide useful creative authority without exposing internal implementation, whereas an accessible binary may remain difficult to understand or maintain. For the same reason, an online requirement, remote execution, and browser delivery should not be treated as synonyms. A browser can execute locally delivered code, while a streamed game can expose sanctioned creator tools. These distinctions undermine technological determinism in both optimistic and pessimistic forecasts. Control must be assessed through actual interfaces, permissions, and dependencies rather than inferred from a distribution label.

The third result is that verification becomes a potential bottleneck as the cost of proposing implementations falls. This is a conditional claim about workflow organization, not a measured productivity result. When a system can produce many plausible transformations, the scarce activity may become identifying which transformations preserve desired behavior and contribute to a coherent experience. Compilation, selected behavioral tests, extended playtesting, and design evaluation provide different kinds of evidence. Their requirements should be specified before a workflow is judged successful. For preservation, unintended changes may count as failures even when players enjoy them. For creative transformation, some departures from the source may be the objective. A credible project must therefore identify both its invariants and its intended differences. Without that distinction, fidelity claims and novelty claims can be used interchangeably to excuse whichever aspect of a reconstruction has not been demonstrated.

Contemporary Cases and the Limits of Acceleration Claims

P. E. Cooper: Memes, Wii Sports, and the Circulation of Playable Hybrids

P. E. Cooper's crossover series provides a practitioner-centered case of game melding as public creative experimentation. In a September 29 post, Cooper attributed a browser version of the

original World at War Zombies map to AI assistance, naming Opus 5.5 and Astra (Cooper, 2026a). The October 1 post framed a subsequent experiment as a World at War and Fortnite combination; the October 2 post announced a Wii Sports crossover (Cooper, 2026b, 2026c). Warby's (2026) contemporaneous report identifies bowling balls, boxing gloves, and a baseball bat among the Wii Sports-inspired additions. It also describes Portal-style portals, Fortnite-style construction, and Mario Kart-inspired driving with banana hazards. These details provide concrete examples of the trend without establishing identical implementation across its demonstrations. The supplied bowling still adds direct visual evidence of the avatar, ball, and scoring overlay, without independently verifying the underlying reconstruction of Wii Sports. The series makes a broader conceptual change visible: participants can propose combinations organized around recognizable actions, not only recognizable appearances.

The combinations examined here draw much of their immediate legibility from familiar franchises and recognizable mechanics. That observation concerns the selected cases, not a measured distribution of all melding activity. Their visibility should not define the limits of what the practice might become. Less familiar games could contribute unusual movement systems, interfaces, resource rules, or narrative structures to combinations that are not primarily organized around recognition. Such work might develop into a distinct mode of user-generated gaming in which players specify relationships between systems rather than request another variation of an established title. Its novelty would depend on the resulting play, not merely on the obscurity of its ingredients. Reduced implementation effort could make these experiments more feasible while leaving coherence and discoverability unresolved. The transition from familiar crossover spectacle to an enduring creative medium is therefore a possibility for investigation rather than an outcome established by the present examples.

Cooper's meme-based Zombies example foregrounds a form of melding that operates through interface and cultural reference rather than an immediately identifiable transfer of another game's complete rule system (Cooper, n.d.; Figure 1). The supplied still shows a smartphone messaging interface held in front of a zombie in the World at War setting. The phone display contains an image of the nearby zombie, creating a visual loop between the game world and the screen presented within it. This juxtaposition places a familiar contemporary communication interface inside an otherwise hostile survival scene. Read alongside the supplied description of fighting zombies with memes, the image suggests how modders can recast the tone and expected repertoire of a game. Its novelty is not reducible to graphical realism or the volume of newly generated content. Instead, the incongruity makes the interface itself part of the joke and the player's apparent toolset. A still cannot establish whether sending an image produces damage, changes enemy behavior, or merely supplies a visual effect. The example therefore identifies an interface-level and memetic recombination while leaving its precise executable consequences open to verification.

Figure 1 Meme-Based Interface Remixing in World at War Zombies



P. E. Cooper (@p_e_cooper), World at War Zombies meme demonstration. The phone display introduces a messaging interface and an image of the nearby zombie into the survival-horror scene.

The Wii Sports bowling and Call of Duty combination illustrates why transferred actions deserve separate analysis from imported assets (Figure 2). The supplied still visibly combines a bowling avatar and ball with a power meter, a “SPARE” feedback label, and the surrounding Zombies interface. A bowling ball can function as a visual reference, but implementing rolling, collision, targeting, or knockdown would also reorganize the player’s available actions. The relevant research question is which of those relationships a particular hybrid actually implements, not whether its imagery evokes both franchises. Similarly, a Portal-inspired tool could change spatial navigation, while a Fortnite-inspired building

action could change defensive planning. These are analytical possibilities suggested by the reported examples, not verified findings about Cooper’s underlying code. A hybrid need not reproduce the complete rules of either source game to establish a distinct play proposition. Evaluation should therefore compare operational affordances and resulting play rather than count recognizable properties. This distinction connects the contemporary demonstrations to earlier computational blending research while preserving the difference between a compelling concept and a technically validated integration.

Figure 2 Wii Sports Bowling in World at War Zombies



P. E. Cooper, Wii Sports crossover (Cooper, 2026c). A bowling avatar, ball, power meter, and “SPARE” feedback appear within the Zombies environment, juxtaposing sports-game presentation with survival combat. [Wii Sports crossover post](#)

Cooper’s posts also expose an evidentiary asymmetry between visible experimentation and the hidden development process. The phrase “pull the source code” expresses the creator’s account but does not specify whether the workflow reconstructed compiled functions, adapted existing code, rebuilt observed behavior, or combined these routes (Cooper, 2026a). The cited posts do not supply a controlled duration comparison, complete provenance record, or independently assessed fidelity measure.

Warby (2026) reported that the crossovers were not then available as readily accessible public mods, limiting that report’s basis for independent playtesting. Neither public attention nor a familiar scene resolves these uncertainties. A reproducible case study would require versioned implementation records, dependency disclosures, tests, and an account of human intervention. The lack of those materials in the reviewed sources does not invalidate the creative demonstrations; it limits what researchers can infer from them. The case supports examining a new mode of rapid

participatory experimentation without treating social-media visibility as evidence that entire commercial games have become interchangeable.

The cultural expectation of conversational creation predates these demonstrations. Spike Jonze's *Her* (2013) situates the game designer Amy's work on the fictional Perfect Mom game alongside conversation with an operating system. The screenplay describes her speaking with Ellie while working on the game, linking creative activity to a conversational digital companion (Jonze, 2013). Read as a cultural analogy, this scene helps articulate the attraction of expressing an idea in ordinary language while software mediates its realization. It is not technical evidence of autonomous development, nor does the cited passage establish a specific source-code generation workflow. Its relevance is the imagined reduction of distance between intention and a responsive creative environment. Contemporary coding agents make that aspiration a practical research question. Whether conversation produces an intelligible, maintainable game remains distinct from whether it makes creation feel immediate.

Reconstruction, Rewriting, and Runtime Bridges

One detailed developer account complicates the popular claim that complete games are now routinely decompiled overnight. Heumann (2026) described a three-month project involving multiple coding-agent tools, including Claude Code and Codex CLI, rather than a single unattended session. He reported reconstruction of 99% of functions, with 83% byte-exact, and estimated total usage at 600–700 billion tokens. He also stated that parts of the logs were lost and that the reconstructed code remained private. These are self-reported project outcomes, not independently reproduced benchmarks. The account's analytical importance is therefore narrower than a universal speed claim: it illustrates the potential scale of agent-supported reconstruction and the continuing importance of validation. Its success criteria also concern different levels of completeness that should not be collapsed. Function coverage does not automatically measure gameplay fidelity, while byte matching addresses a stronger but still specifically defined reconstruction objective. The case supports investigating acceleration, but not replacing a missing

controlled baseline with a rhetorical comparison between one night and several years.

The Rust mashup associated with chasmlol illustrates a different route to recombination. Its documentation describes an IW4L Rust rewrite associated with Modern Warfare 2 and the combination of elements connected to Skate 3 and Minecraft (chasmlol, 2026a). That description should not be converted into a claim that three original source repositories were recovered and seamlessly fused. Related Skate 3 engine documentation explicitly credits more than two years of prior reverse-engineering work and identifies continuing limits to gameplay parity (SK8-ENGINE, 2026). Such acknowledgments are consequential evidence, not incidental footnotes. They show why the final integration interval and the total knowledge-production interval must be distinguished. A rapid new result can depend on a mature interpretive infrastructure assembled by other people. Recognizing this does not diminish the integrator's contribution or deny that an agent may accelerate work. It changes the attribution and evaluation question from who produced the visible demonstration to which accumulated capabilities made that demonstration possible.

The user-supplied compilation frame credited on screen to @Chasmmm offers a visible counterpart to this discussion of cross-game integration (Figure 3). It places a realistically rendered first-person firearm and hands within Minecraft's block-based forest, alongside a minimap, item slots, and an ammunition indicator. The combination makes the different representational conventions legible in a single view. Read as a Call of Duty and Minecraft hybrid, the still suggests how a familiar combat interface can be relocated into a world associated with construction and resource manipulation. It does not, however, reveal whether the implementation imports an asset, reconstructs weapon behavior, or connects separate runtimes. The source credit identifies the creator attribution carried by the reel, whereas the supplied X link identifies @snelstack as the account circulating the compilation (snelstack, 2026). This distinction prevents an uploader credit from becoming an unsupported claim about every contribution to the mod. Together, the three figures locate contemporary melding at the levels of interface, game feedback, and cross-world presentation while reserving architectural conclusions for the accompanying technical documentation.

Figure 3 Call of Duty-Style Weaponry in a Minecraft Environment



@snelstack (2026); the visible source credit reads “@Chasmmm.” The frame combines Call of Duty-style first-person weaponry and interface elements with a Minecraft environment. Creator attribution and compilation circulation are recorded separately. Game-melding compilation

SkyCraft supplies the clearest architectural contrast because its repository describes two running games connected through an SKSE plugin, a Fabric mod, and shared memory (chasmlol, 2026b). The documentation assigns Minecraft a substantial role in mechanics while Skyrim supplies its world and presentation; it does not claim that both engines have been replaced. This makes the project especially useful for distinguishing interoperability from decompilation. A player may perceive a unified hybrid while the underlying software remains distributed across separate processes. The analytical consequence is that the boundary of a game becomes a question of coordinated experience rather than a requirement that everything reside in one engine. The arrangement introduces dependencies on both host systems and their changing environments. A successful bridge can therefore expand expressive possibilities while increasing maintenance obligations. The case demonstrates why evidence of a compelling combination should be accompanied by an account of the architecture that sustains it.

Taken together, these cases support a more defensible concept of acceleration than the overnight narrative. Table 2 records their differing evidentiary roles alongside the dated 2025–2026 developments introduced earlier. Acceleration can concern interpretation, implementation, porting, testing, or the coordination of existing components, and gains at one stage need not determine total project duration. A comparison should include prior tools, available documentation, human supervision, computing budgets, failed attempts, and later maintenance. It should also identify whether the target was a demonstration, a playable prototype, a faithful reconstruction, or a distributable product. The public cases are valuable because they establish questions that controlled studies can now investigate. They are insufficient for estimating how frequently such projects succeed across unfamiliar games, ordinary budgets, or less favorable conditions. The justified conclusion is that executable recombination deserves systematic study, not that conventional game production has already become obsolete.

Table 2 Selected 2025–2026 Developments and Their Evidentiary Limits

Date (2026 unless noted)	Development and source	Bounded evidentiary contribution
May 16, 2025	Conversational Darth Vader in Fortnite (StarWars.com Team, 2025; Farokhmanesh, 2025)	Official AI character; player-induced profanity exposed guardrail limits, not source-code access.
January 29	Project Genie (Rivas et al., 2026)	Experimental interactive-world creation; not proof of complete game generation.
February 4	Roblox Cube (Roblox, 2026)	Interactive-object generation beta; not a complete autonomous production pipeline.
June 17–30	PUBG Ally beta (Burnes, 2026)	Time-limited AI-teammate deployment; not evidence of universal availability.
September 4	Decompile-Diverge preprint (Liu et al., 2026)	Additional behavioral checks can expose divergence despite supplied-test success.
September 29	Cooper's browser-port account (Cooper, 2026a)	Creator attribution to AI tools; no independent reconstruction or duration benchmark.
October 1–2	Fortnite and Wii Sports crossovers (Cooper, 2026b, 2026c; Warby, 2026)	Creator framing and reported action recombination, including bowling; not verified source recovery or a tested public release.
October 9	FPS reconstruction account (Heumann, 2026)	Three-month self-report; private implementation and incomplete logs limit replication.
October 13	Xbox Labs announcement (Asha, 2026)	Institutional interest in experimentation; not an unrestricted remix license.
October 15 snapshot	Rust mashup and related Skate 3 engine documentation (chasmlol, 2026a; SK8-ENGINE, 2026)	Rewrite and integration claims; explicitly inherited reverse-engineering work.
October 15 snapshot	SkyCraft documentation (chasmlol, 2026b)	Runtime-bridge architecture; not two completely decompiled games.

Discussion: Competing Industry Futures

Bespoke Play Without Universal Replacement

One plausible future is an expanded ecology of bespoke games in which players use AI assistance to specify combinations that established productions do not offer. This could include small-group experiences, alternative control schemes, revised pacing, or

unusual intersections of mechanics. The scenario follows from reduced implementation friction, not from an assumption that every player wishes to become a designer. Choosing among generated alternatives still requires time, judgment, and a vocabulary for articulating preferences. Some players may welcome those responsibilities, while others may prefer the surprise and coherence of another person's completed work. Shared games also provide stable reference points for discussion,

competition, and collective memory that unlimited personalization might complicate. Consequently, bespoke play should be understood as a potential expansion of available forms rather than the inevitable replacement of authored games. Its success would depend on whether the experience of commissioning, adjusting, and maintaining a personal game is itself rewarding. The relevant empirical question is how many players adopt that role, for which purposes, and for how long.

Personalization also raises the question of whether a game should satisfy existing preferences or challenge them. A system instructed to maximize immediate satisfaction might repeatedly select familiar structures, whereas a designer may intentionally introduce friction, ambiguity, or an unfamiliar perspective. This is a conceptual tension, not a claim about the behavior of every current model. The distinction matters because bespoke generation could increase choice without increasing the range of experiences a person encounters. A useful evaluation would separate preference matching from discovery, skill development, and sustained appreciation. It would also preserve a player's ability to reject personalization or share a stable configuration with others. Under this scenario, the strongest creator tools would not simply produce more options. They would make the consequences of those options understandable and support deliberate commitment to an experience. Bespoke gaming would then represent a new relationship between design and play, not merely an automated recommendation system that happens to generate software.

The implications of bespoke play could extend beyond the gaming industry to the organization of media consumption more broadly. Consider a hypothetical, rights-cleared streaming service resembling Netflix that offers both the original episodes of *The Office* and a premium option for prompting personalized episodes. This is an illustrative business scenario, not a claim about a current service or licensing agreement. An audience member might specify a different character pairing, situation, or ending while the platform supplies permitted narrative and audiovisual materials. The central transaction would shift from choosing among completed works toward commissioning variations within an authorized fictional world. An industry that accepts player-directed game recombination could provide other media industries with a model for organizing that participation. Audiences would gain greater influence over what they consume, while platforms could acquire greater control over the permitted range of that influence. The relevant question is therefore not only whether consumers become creators, but which institutions define the conditions of their authorship.

Personalized media could nevertheless weaken the shared reference points through which audiences recognize one another as a public. Under this scenario, two people could follow the same franchise while encountering different episodes, character developments, or endings. Discussion might shift from interpreting a common work to comparing experiences that no longer share a stable narrative baseline. This fragmentation could complicate the fandom economy: coordinated releases, recognizable moments, fan discussion, and shared cultural attachments help connect attention to ongoing commercial participation. It could also produce social disconnection when an individually satisfying version becomes difficult to discuss with others. These are conditional risks, not established effects of present game-melding projects. Shared prompt collections, publicly available variants, and clearly distinguished canonical releases could instead create new forms of

collective participation. Whether audience-directed media fragments fandom or reorganizes it should be evaluated through community continuity and shared engagement, not inferred from the volume of personalized content alone.

Abundance, Clones, and the Meaning of Quality

A second scenario concerns content abundance and the circulation of repetitive, unstable, or opportunistic releases. The colloquial term *slop* captures an anxiety about such abundance but does not provide a defensible research measure. This article instead proposes evaluating coherence, originality, technical stability, accessibility, provenance, and continued support as distinct dimensions. A generated game might perform well on one dimension and poorly on another, just as a conventionally produced game might. Clone status should likewise distinguish lawful similarity in general mechanics from unauthorized copying of protected expression, a distinction grounded in copyright's separation of ideas and expression (Copyright Act, 17 U.S.C. § 102). The economic concern is not only that inexpensive production could increase supply. It is that discoverability, evaluation, and trust might fail to improve at the same pace. Under those conditions, attention could become harder to allocate even while technically capable creators gain new opportunities. The outcome would depend on distribution incentives and curation, rather than following automatically from AI use.

This abundance scenario creates an institutional role for quality signals, but it also creates opportunities for new forms of gatekeeping. A platform could require provenance records, functional checks, meaningful descriptions, and support commitments while still allowing experimental work. Alternatively, it could use opaque criteria that favor established publishers or penalize unfamiliar forms. These possibilities should be assessed through observable practices rather than assumed intentions. Useful indicators include the rate at which users encounter duplicate releases, the survival of independently maintained projects, and the relationship between visibility and documented quality. Researchers should also examine whether curation rewards novelty at the expense of reliability or rewards superficial polish at the expense of deeper design. The relevant comparison is not between unlimited release and perfect centralized judgment. It is between imperfect governance arrangements that distribute different burdens among creators, platforms, and players. An increase in generated output therefore makes the politics of discovery more important, not less.

Enclosure and the Governed Hybrid

A third scenario is stronger enclosure, including greater reliance on server-authoritative systems or remote execution. This could reduce access to the implementation details that some forms of independent modification require. However, the causal claim that AI-assisted modding will force an industry-wide transition to streaming is not established by the evidence reviewed here. Publishers may weigh modification risks against performance requirements, infrastructure costs, audience preferences, preservation commitments, and the value of creator communities. They may also distinguish competitive integrity from single-player experimentation rather than impose one rule across an entire catalog. Remote execution can constrain unauthorized inspection while still supporting a licensed creation interface. Locally installed software can likewise impose substantial restrictions through contracts, dependencies, and distribution policies. The

scenario should therefore be studied through changes in practical modification rights and capabilities, not simply through the number of games labeled online. Enclosure is a possible strategic response, but its extent and consequences remain contingent.

These concerns intersect with the shift toward digital distribution and service-dependent play, where purchasing access need not secure continuing control over an executable work. Sony's July 1, 2026 announcement provides a concrete example of a further move away from physical distribution. It states that physical disc production for new games releasing on PlayStation consoles will end in January 2028, while excluding games already released or released on disc before that date from the change (Shuman, 2026). The announcement concerns distribution format; it does not establish that every affected game will require online execution. For collectors and preservationists, the consequential distinction is between retaining a copy and retaining the conditions necessary to use it. A locally stored game can still depend on authentication, servers, or other resources outside the purchaser's control. Stronger restrictions introduced in response to AI-assisted modification could deepen that dependence, although the reviewed evidence does not establish such a causal response. The preservation question is thus whether changing distribution and

access arrangements leave practical means to maintain games after commercial support ends.

A governed hybrid is already visible as an announced direction rather than an established outcome. On October 13, Xbox introduced Xbox Labs in a statement that explicitly discussed recent decompilation, porting, and remixing experiments and described opportunities for greater access and co-creation (Asha, 2026). The same statement reaffirmed responsibility for intellectual property and existing communities. It did not grant blanket permission to redistribute games or guarantee a particular licensing framework. Nevertheless, the announcement is significant counterevidence to the assumption that incumbents can respond only by closing access. It suggests that experimentation and protection can be articulated within the same strategic position. Whether that position yields meaningful creator autonomy depends on the tools, permissions, economic terms, and preservation arrangements that follow. Table 3 treats this hybrid alongside bespoke, abundance-driven, and enclosure scenarios without assigning probabilities. Their coexistence across different genres, platforms, and rights holders is more analytically plausible than a universal outcome inferred from a small set of demonstrations.

Table 3 Conditional Industry Scenarios and Observable Indicators

Scenario	Potential benefit and risk	Indicators to investigate
Bespoke participation	Closer fit to local interests; possible loss of shared reference points or burdensome customization.	Sustained creator participation, repeat play, configuration sharing, and opt-out behavior.
Abundance and curation	Lower publication barriers; possible duplication, weak support, and attention congestion.	Duplicate exposure, independent-project visibility, stability, and support persistence.
Stronger enclosure	Greater control over execution and distribution; reduced independent modification or preservation access.	Changes in interfaces, local capabilities, permissions, shutdown provisions, and creator exit options.
Governed experimentation	Licensed collaboration and clearer responsibilities; possible platform dependence or selective access.	Actual tool releases, license terms, revenue allocation, appeals, portability, and archival provisions.

Governance, Authorship, and Sustainable Participation

Governance begins by distinguishing copyrightability from permission, because the two answer different questions. The U.S. Copyright Office's report on AI and copyrightability explains that human-authored expression and qualifying human selection or arrangement may remain protected even when AI tools are involved (U.S. Copyright Office, 2025). That principle does not confer permission to reuse someone else's protected code, artwork, music, dialogue, or characters. Nor does the absence of protection for a purely generated element extinguish rights in material incorporated from another work. A melded project therefore requires analysis at the level of its constituent contributions and transformations. Documentation should identify what was independently created, what was generated, what was adapted, and what remains dependent on a separately licensed game. The goal is not to presume that every combination is prohibited. It is to prevent an argument about human authorship from being mistaken for a

blanket authorization to copy. This distinction is particularly important when a project's own source-code license is more permissive than the licenses governing its required assets.

Reverse engineering requires a similarly careful separation of technical ability and legal entitlement. United States law provides a limited interoperability-related exception in 17 U.S.C. § 1201(f), while other provisions govern reproduction, adaptation, and fair use (Copyright Act, 17 U.S.C. §§ 106–107, 1201). These provisions do not establish that every decompilation, access-control circumvention, or redistribution is lawful. Their application depends on purpose, conduct, jurisdiction, and the relevant facts. Consequently, owning a copy of a game should not be described as possessing unrestricted rights to distribute its assets or reconstructed implementation. A research framework should record the permission basis claimed by a project without presenting that claim as a judicial determination. It should also distinguish preservation, private experimentation, interoperability, public distribution, and commercial exploitation as potentially different

contexts. This article offers an analytical framework rather than legal advice about any identified repository. Responsible publication requires qualified legal assessment where a proposed use depends on contested rights or exceptions.

Provenance should also preserve the social history of a project, not merely the name of the latest model. A useful record would connect source versions, tools, dependencies, authors, permissions, and consequential design decisions. It would identify prior reverse-engineering work and document the boundaries of an agent's contribution. Such a record could help maintainers understand failures and help contributors receive credit when a visually novel integration attracts disproportionate attention. It could also distinguish a preservation build from a creative reinterpretation by recording intentional departures from the source. These are proposed governance functions rather than demonstrated effects of a particular metadata standard. Their value depends on whether records remain intelligible, affordable to maintain, and connected to real decisions. Excessive documentation burdens could exclude small creators, so requirements should scale with distribution risk and claimed fidelity. The objective is accountable participation, not an administrative threshold that only large studios can satisfy.

Treating video games as an art form makes the loss of playable access a cultural issue as well as a consumer concern. Ubisoft's service-closure notice for *The Crew* states that its servers would shut down on March 31, 2024 and that the game would thereafter be inaccessible through that service (Ubisoft, n.d.). Firewalk likewise announced that *Concord* would go offline on September 6, 2024, with refunds covering both digital and physical purchases (Ellis, 2024). These examples show why a purchased copy, including a disc, does not necessarily preserve an online-dependent work. Retaining images and recordings can document its appearance without retaining the rules and interactions through which it was experienced. Online-only architectures, authentication requirements, and access controls can consequently obstruct preservation even when the underlying files survive. That effect should be distinguished from an unsupported claim that every such restriction was introduced specifically to prevent archiving. The cultural value of preserving playable histories supports examining whether rights holders provide offline modes, server tools, or other authorized continuity arrangements when support ends.

Preservation makes the distinction between modification and stewardship particularly consequential. A reconstructed game might become easier to execute while becoming a less reliable record of its historical behavior. A preservation-oriented project should therefore retain a reference description, document uncertain reconstructions, and separate intentional improvements from the archival target. An accessibility adaptation or compatibility repair can be valuable without being presented as an unchanged original. Similarly, an artistically successful hybrid can deserve preservation as a new cultural artifact rather than as a substitute for its source games. These distinctions create different requirements for documentation, versioning, and comparison. They also make room for collaboration between technical maintainers and scholars concerned with historical context. The purpose is not to freeze games against all transformation, but to preserve the ability to distinguish transformations from the works they reinterpret. That distinction supports both experimentation and credible historical inquiry.

Distribution policies offer one practical site for implementing this distinction between enabling creativity and preserving accountability. Steam's Content Survey differentiates relevant AI-generated content and asks developers to address matters associated with its inclusion, including additional concerns for live generation (Valve, n.d.). This is an example of platform governance, not evidence that disclosure alone guarantees quality or resolves ownership. A stronger evaluative approach would connect disclosure to intelligible information about function, dependencies, and support. Players need to understand whether a release is experimental, what games or services it requires, and which limitations remain. Developers likewise need predictable rules and a process for contesting erroneous decisions. These needs become more pressing when an update to one component can destabilize a combination maintained by someone else. A platform that supports creator participation must therefore consider maintenance relationships alongside initial publication. Otherwise, it may encourage the production of impressive fragments without supporting the conditions necessary for durable games.

Research Agenda and Implications

A credible research agenda should begin with paired comparisons that make acceleration claims interpretable. Studies could assign comparable reconstruction or interoperability tasks to human-only, AI-assisted, and agent-led workflows while recording expertise, prior tooling, supervision, and computational expenditure. A common budget would be necessary, although monetary cost, elapsed time, and human effort should remain separate outcomes. Evaluators should specify whether a task requires behavioral equivalence, a playable prototype, or an intentionally transformed mechanic. Benchmarks should include unfamiliar projects and dependencies rather than relying exclusively on well-documented, culturally prominent games. Independent tests should examine persistence, transitions, and unusual states as well as ordinary execution. Negative results and abandoned attempts should be retained to reduce the selection bias of public demonstrations. Such designs would make it possible to identify where assistance changes work rather than merely observing that a striking result was produced. They would also clarify when apparent savings are offset by verification or maintenance demands. This would make the evidence useful to teams deciding whether a demonstrated capability transfers to their own projects, rather than only to audiences evaluating a spectacular result.

A complementary program should evaluate the human experience of melding rather than treating technical success as sufficient. Blinded playtests could compare hybrids with appropriate baselines on coherence, learnability, enjoyment, accessibility, and perceived novelty. Longitudinal studies could examine whether initial enthusiasm persists after repeated play and after the novelty of the combination declines. Creator studies should document learning, ownership of decisions, confidence, and the ability to diagnose or revise generated systems. Community studies could examine attribution disputes, mentoring practices, collaboration across experience levels, and the distribution of maintenance work. Platform research should connect policies to actual publication, discoverability, removal, and compensation outcomes instead of relying only on announced intentions. These strands would allow researchers to distinguish participation that is merely nominal from participation that changes practical creative authority. They would also reveal whether AI-assisted production

broadens whose experiences are represented or reproduces a narrower set of familiar templates. Table 4 translates these

proposed investigations into observable measures and necessary cautions.

Table 4 Research Questions, Proposed Measures, and Interpretive Safeguards

Research question	Proposed evidence	Required safeguard
Where does AI accelerate work?	Elapsed time, human hours, compute, failure and rework logs.	Match task scope, prior knowledge, budget, and completion criteria.
Does reconstruction preserve behavior?	Independent differential tests, state-transition coverage, persistence checks.	Keep evaluator-controlled checks separate from the generating agent.
Does melding improve play?	Blinded comparisons, repeat-play studies, accessibility and coherence assessments.	Distinguish novelty effects from sustained experience.
Who gains creative authority?	Decision histories, attribution, interviews, mentoring and maintenance networks.	Record inherited labor and include abandoned or unsuccessful projects.
How do platforms shape outcomes?	Publication, discovery, removal, licensing, compensation, and portability records.	Separate policy announcements from implementation and observed effects.

The resulting implications extend to how developers, educators, and preservation institutions organize responsibility. Developers could treat compatibility interfaces and clearly licensed components as strategic design choices rather than responses made only after unauthorized experimentation appears. Educators could teach students to document assumptions, construct independent tests, and analyze rights alongside learning to prompt or integrate an agent. Preservation institutions could distinguish faithful reference builds from creative derivatives, preserving both when appropriate without confusing their purposes. In each setting, generated code should enter a controlled workflow with bounded permissions, review, and a recoverable history. These recommendations concern the safe organization of development, not techniques for bypassing protections. Their common principle is that increased generation capacity should be matched by increased capacity to evaluate and govern what has been generated. The central educational shift is therefore from celebrating output alone to developing judgment about systems, evidence, and consequences. That judgment remains necessary whether the final experience is bespoke, commercially authored, collaboratively modified, or remotely delivered.

Conclusion

AI game melding marks a consequential convergence, but its significance becomes clearer when the strongest claims are narrowed. Computational game blending is not new, and public demonstrations do not establish that complex games can routinely be reconstructed overnight without substantial inherited knowledge or supervision. The emerging development is the widening practical connection between AI-assisted interpretation, executable reconstruction, runtime interoperability, and participatory design. That connection could enable forms of play that would otherwise remain prohibitively difficult to implement. It could also generate unstable combinations, obscure prior labor, complicate rights, and place new power in the hands of infrastructure providers. The evidence reviewed here supports taking these possibilities seriously without presenting any one of them as an accomplished industry

transformation. It also supports a shift in research focus from isolated output quality to the systems that sustain a playable, maintainable, and legitimately distributed work. The proposed composability stack provides one way to make that shift analytically explicit.

The decisive question is therefore not whether AI will make every game user-generated or persuade every publisher to conceal its software behind streaming. Cooper's crossovers make a more precise question tangible: which forms of creative authority remain accessible as reconstruction and recombination become easier? Bespoke creation, curated abundance, stronger enclosure, and licensed experimentation may develop simultaneously, with different consequences across communities and genres. What matters is whether players and creators can understand the systems they modify, preserve meaningful choices, receive appropriate credit, and maintain the results. An industry that supports those conditions could gain new creative partners without abandoning authored experiences or intellectual-property stewardship. An industry that responds only through opaque restrictions or unaccountable volume could reduce the cultural value of the same technical capabilities. AI game melding thus makes the governance of programmable play newly urgent. The future remains open not because technical change is inconsequential, but because its most important ramifications depend on choices about verification, access, authorship, and participation that technology alone cannot resolve.

References

1. Alharthi, S. A. (2025). Generative AI in game design: Enhancing creativity or constraining innovation? *Journal of Intelligence*, 13(6), Article 60. <https://doi.org/10.3390/jintelligence13060060>
2. Asha. (2026, October 13). *Introducing Xbox Labs*. Xbox Wire. <https://news.xbox.com/en-us/2026/10/13/introducing-xbox-labs/>

3. 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), Article 2568. <https://doi.org/10.54517/m.v5i2.2568>
4. 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), 1–14. <https://doi.org/10.58567/jie03010001>
5. Bohemia Interactive. (2015, March 26). *Make Arma Not War winners*. <https://www.bohemia.net/en/blog/make-arma-not-war-winners>
6. Burnes, A. (2026, June 17). *PUBG Ally duo mode two-week beta available now, adding a collaborative AI teammate to your squad*. NVIDIA. <https://www.nvidia.com/en-us/geforce/news/pubg-ally-ai-teammate-beta-available-now/>
7. chasmlol. (2026a). *2010-rust-rewrite-mashup* [Computer software and project documentation]. GitHub. Retrieved October 15, 2026, from <https://github.com/chasmlol/2010-rust-rewrite-mashup>
8. chasmlol. (2026b). *SkyCraft* [Computer software and project documentation]. GitHub. Retrieved October 15, 2026, from <https://github.com/chasmlol/SkyCraft>
9. Cooper, P. E. [@p_e_cooper]. (n.d.). *Posts* [X profile]. X. Retrieved October 15, 2026, from https://x.com/p_e_cooper
10. Cooper, P. E. [@p_e_cooper]. (2026a, September 29). *[Video post describing AI-assisted adaptation of World at War Zombies for a web browser]*. X. https://x.com/p_e_cooper/status/2104980433972703483
11. Cooper, P. E. [@p_e_cooper]. (2026b, October 1). *[Video post presenting a World at War Zombies and Fortnite crossover]*. X. https://x.com/p_e_cooper/status/2105524743902208317
12. Cooper, P. E. [@p_e_cooper]. (2026c, October 2). *[Video post presenting a Wii Sports and World at War Zombies crossover]*. X. https://x.com/p_e_cooper/status/2105822578728436169
13. Copyright Act, 17 U.S.C. §§ 102, 106–107, 1201. <https://www.copyright.gov/title17/>
14. Ellis, R. (2024, September 3). *An important update on Concord*. PlayStation.Blog. <https://blog.playstation.com/2024/09/03/an-important-update-on-concord/>
15. Farokhmanesh, M. (2025, May 16). *Fortnite players are already making AI Darth Vader swear*. WIRED. <https://www.wired.com/story/fortnite-darth-vader-slurs-swears-generative-ai/>
16. Francis, B. (2017, May 3). *Some advice for modders from the creator of Playerunknown's Battlegrounds*. Game Developer. <https://www.gamedeveloper.com/design/some-advice-for-modders-from-the-creator-of-i-playerunknown-s-battlegrounds-i->
17. 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/>
18. Geheeb, J., Schwarz, M. J., Dyrda, D., & Groh, G. (2026). LLMs are the ideal candidate for mixed-initiative game design pillar workflows. In *Proceedings of the 21st International Conference on the Foundations of Digital Games*. Association for Computing Machinery. <https://doi.org/10.1145/3815598.3815653>
19. Guinness World Records. (n.d.). *Best-selling videogame modification*. <https://www.guinnessworldrecords.com/world-records/87935-best-selling-videogame-modification>
20. Guzdial, M., & Riedl, M. (2018). Automated game design via conceptual expansion. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 14(1), 31–37. <https://doi.org/10.1609/aiide.v14i1.13022>
21. Heumann, M. (2026, October 9). *500+ billion tokens later: Letting AI agents decompile a first-person shooter*. Maurice's Blog. <https://momo5502.com/posts/2026-10-09-game-decompilation/>
22. Jonze, S. (2013). *Her* [Screenplay]. The Internet Movie Script Database. <https://imsdb.com/scripts/Her.html>
23. Kim, S. (2018, June 27). *Copyright lawsuit dropped against Fortnite creators, ending legal battle*. Bloomberg. <https://www.bloomberg.com/news/articles/2018-06-27/pubg-drops-lawsuit-against-creators-of-gaming-phenom-fortnite>
24. Liu, C., Raff, E., & Micinski, K. (2026). *When LLM decompilers recompile more and preserve less* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2609.05370>
25. Mescon, S. (2009, March 19). *Postmortem: Defense of the Ancients*. Game Developer. <https://www.gamedeveloper.com/design/postmortem-i-defense-of-the-ancients-i->
26. Mustard, D. (2017, September 12). *Fortnite Battle Royale launches September 26*. PlayStation.Blog. <https://blog.playstation.com/2017/09/12/fortnite-battle-royale-launches-september-26/>
27. Nieborg, D. B., & van der Graaf, S. (2008). The mod industries? The industrial logic of non-market game production. *European Journal of Cultural Studies*, 11(2), 177–195. <https://doi.org/10.1177/1367549407088331>
28. Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (pp. 1–22). Association for Computing Machinery. <https://doi.org/10.1145/3586183.3606763>
29. Ratican, J., & Hutson, J. (2024a). Adaptive worlds: Generative AI in game design and future of gaming, and interactive media. *ISRG Journal of Arts, Humanities and*

- Social Sciences*, 2(5), 284–290. <https://doi.org/10.5281/zenodo.13894497>
30. Ratican, J., & Hutson, J. (2024b). Video game development 3.0: AI-driven collaborative co-creation. *Metaverse*, 5(2), Article 2904. <https://doi.org/10.54517/m2904>
31. Rivas, D., Breece, E., & Chambers, S. (2026, January 29). *Project Genie: Experimenting with infinite, interactive worlds*. Google. <https://blog.google/innovation-and-ai/models-and-research/google-deepmind/project-genie/>
32. Robinson, A. (2021, November 27). *Zelda 64 has been fully decompiled, potentially opening the door for mods and ports*. Video Games Chronicle. <https://www.videogameschronicle.com/news/zelda-64-has-been-fully-decompiled-potentially-opening-the-door-for-mods-and-ports/>
33. Roblox. (2026, February 4). *Accelerating creation, powered by Roblox's Cube Foundation Model*. <https://about.roblox.com/newsroom/2026/02/accelerating-creation-powered-roblox-cube-foundation-model>
34. Sarkar, A., & Cooper, S. (2022). *Latent combinational game design* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2206.14203>
35. Scacchi, W. (2010). Computer game mods, modders, modding, and the mod scene. *First Monday*, 15(5). <https://doi.org/10.5210/fm.v15i5.2965>
36. Schreier, J. (2010, October 13). Valve revives Defense of the Ancients RTS in 2011. *WIRED*. <https://www.wired.com/2010/10/dota-2/>
37. Shuman, S. (2026, July 1). *Physical disc production ending in January 2028 for new games releasing on PlayStation consoles*. PlayStation.Blog. <https://blog.playstation.com/2026/07/01/physical-disc-production-ending-in-january-2028-for-new-games-releasing-on-playstation-consoles/>
38. SK8-ENGINE. (2026). *skate-3-rust-engine* [Computer software and project documentation]. GitHub. Retrieved October 15, 2026, from <https://github.com/SK8-ENGINE/skate-3-rust-engine>
39. snelstack [@snelstack]. (2026, October 13). [Video compilation of game-melding demonstrations with on-screen source credit to @Chasmmm]. X. <https://x.com/snelstack/status/2110002978744606773>
40. Sotamaa, O. (2010). When the game is not enough: Motivations and practices among computer game modding culture. *Games and Culture*, 5(3), 239–255. <https://doi.org/10.1177/1555412009359765>
41. StarWars.com Team. (2025, May 16). *This will be a day long remembered: Speak with Darth Vader in Fortnite*. StarWars.com. <https://www.starwars.com/news/speak-with-darth-vader-in-fortnite>
42. Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., & Togelius, J. (2018). Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games*, 10(3), 257–270. <https://doi.org/10.1109/TG.2018.2846639>
43. Tan, H., Li, W., Tian, X., Wang, S., Liu, J., Li, J., & Zhang, Y. (2025). *SK²Decompile: LLM-based two-phase binary decompilation from skeleton to skin* (Version 1) [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2509.22114>
44. Tan, H., Li, W., Tian, X., Wang, S., Liu, J., Li, J., & Zhang, Y. (2026). SK²Decompile: LLM-based two-phase binary decompilation from skeleton to skin. In *Proceedings of the International Conference on Learning Representations*. https://proceedings.iclr.cc/paper_files/paper/2026/file/9132583dd869acb0cafeb0ed25eb031a-Paper-Conference.pdf
45. Tan, H., Luo, Q., Li, J., & Zhang, Y. (2024). LLM4Decompile: Decompiling binary code with large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 3473–3487). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.emnlp-main.203>
46. Ternar, A., Denisova, A., Cunha, J. M., Kultima, A., & Guckelsberger, C. (2026). Generative AI in game development: A qualitative research synthesis. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (Article 452, pp. 1–24). Association for Computing Machinery. <https://doi.org/10.1145/3772318.3791206>
47. Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3), 172–186. <https://doi.org/10.1109/TCIAIG.2011.2148116>
48. Ubisoft. (n.d.). *The Crew: Standard edition* [Game storefront and service-closure notice]. Epic Games Store. <https://store.epicgames.com/en-US/product/the-crew/home>
49. U.S. Copyright Office. (2025). *Copyright and artificial intelligence: Part 2. Copyrightability*. <https://www.copyright.gov/ai/Copyright-and-Artificial-Intelligence-Part-2-Copyrightability-Report.pdf>
50. Valevski, D., Leviathan, Y., Arar, M., & Fruchter, S. (2025). Diffusion models are real-time game engines. In *Proceedings of the International Conference on Learning Representations*. <https://openreview.net/forum?id=P8pqeEkn1H>
51. Valve. (n.d.). *Content survey*. Steamworks Documentation. Retrieved October 15, 2026, from <https://partner.steamgames.com/doc/gettingstarted/contentsurvey>
52. Warby, N. (2026, October 2). *Call of Duty fan creates insane Zombies crossovers with Wii Sports, Portal & Fortnite*. Dexerto. <https://www.dexerto.com/call-of-duty/call-of-duty-fan-creates-insane-zombies-crossovers-with-wii-sports-portal-fortnite-3414983/>