User-Generated Content and Editors in Games: A Comprehensive Survey

HAIHAN DUAN, Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen, China and Guangdong-Hong Kong-Macao Joint Laboratory for Emotion Intelligence and Pervasive Computing, Shenzhen, China
YUYUE LIU, Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen, China, Guangdong-Hong Kong-Macao Joint Laboratory for Emotion Intelligence and Pervasive Computing, Shenzhen, China, and The Chinese University of Hong Kong, Shenzhen, China
WEI CAI, School of Engineering and Technology, University of Washington, Tacoma, Washington, USA

User-Generated Content (UGC) refers to any form of content, such as posts and images, created by users rather than by professionals. In recent years, UGC has become an essential part of the evolving video game industry, influencing both game culture and community dynamics. The ability for users to actively contribute to the games they engage with has shifted the landscape of gaming from a one-directional entertainment experience into a collaborative, user-driven ecosystem. Therefore, this growing trend highlights the urgent need for summarizing the current UGC development in game industry. Our conference paper has systematically classified the existing UGC in games and the UGC editors separately into four types. However, the previous survey lacks the depth and precision necessary to capture the wide-ranging and increasingly complex nature of UGC. To this end, as an extension of previous work, this article presents a refined and expanded classification of UGC and UGC editors within video games, offering a more robust and comprehensive framework with representative cases that better reflects the diversity and nuances of contemporary user-generated contributions. Moreover, we provide our insights on the future of UGC, involving application of artificial intelligence, potential ethical considerations, relationship between games, users and communities, game culture, and game genre, and user creative tendencies.

CCS Concepts: • **Information systems** → **Multimedia content creation**; • **Human-centered computing** → *User studies*;

Yuyue Liu is a visiting student at Artificial Intelligence Research Institute, Shenzhen MSU-BIT University and at Guangdong-Hong Kong-Macao Joint Laboratory for Emotion Intelligence and Pervasive Computing.
This work is supported in part by the Key Field Projects of Ordinary Universities in Guangdong Province (No. 2025ZDZX3050); in part by Shenzhen Science and Technology Program (Grant No. JCYJ20210324124205016); in part by Guangdong-Hong Kong-Macao Joint Laboratory for Emotional Intelligence and Pervasive Computing, Artificial Intelligence Research Institute, Shenzhen MSU-BIT University; and in part by Project 61902333 by National Natural Science Foundation of China.
Authors' Contact Information: Haihan Duan, Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen, China and Guangdong-Hong Kong-Macao Joint Laboratory for Emotion Intelligence and Pervasive Computing, Shenzhen, China; e-mail: duanhaihan@smbu.edu.cn; Yuyue Liu, Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen, China, Guangdong-Hong Kong-Macao Joint Laboratory for Emotion Intelligence and Pervasive Computing, Shenzhen, China, and The Chinese University of Hong Kong, Shenzhen, China; e-mail: yuyueliu@link.cuhk.edu.cn; Wei Cai (corresponding author), School of Engineering and Technology, University of Washington, Tacoma, Washington, USA; e-mail: weicaics@uw.edu.

ACM 1551-6865/2026/6-ART180
https://doi.org/10.1145/3798165

Additional Key Words and Phrases: Games, User-Generated Content, Game Editor, Human-Centered Computing

ACM Reference format:
Haihan Duan, Yuyue Liu, and Wei Cai. 2026. User-Generated Content and Editors in Games: A Comprehensive Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 22, 6, Article 180 (June 2026), 36 pages.
https://doi.org/10.1145/3798165

1 Introduction

User-Generated Content (UGC) refers to any form of content, such as text, posts, images, videos, reviews, and other media which created by users or consumers, rather than official developers [64, 67]. In recent years, UGC has become a cornerstone of the modern video game industry, providing players with the tools to create, share, and engage in creative content beyond the confines of traditional game development. This dynamic between players and games not only enhances engagement but also drives the cultural and economic growth of gaming communities [88, 99]. Given its increasing importance in ensuring the long-term sustainability of video games, UGC is a critical area for ongoing research.

The scope of UGC in video games is vast, encompassing everything from user-designed character appearances to modifications (Mods) that alter game code. The most notable examples in the history of video game development are ***Counter-Strike (CS)***[1] and ***Defense of the Ancients (DOTA)***,[2] where *CS* originated from a mod of *Half-Life*,[3] a **First-Person Shooting (FPS)** video game, and *DOTA* was a multiplayer online battle arena mod of *Warcraft III: Reign of Chaos*.[4] UGC enhances players' gaming experiences by promoting greater engagement and giving them a sense of control. Through such user-centered content creations, users can modify game mechanics, such as altering character models or changing gameplay rules with mods, leading to innovative experiences that increase enjoyment and satisfaction [20, 49].

While UGC has been studied in specific fields like brands marketing [12], tourism [92], hotel experiences [4], digital well-being [16], social media [60], and so on, with limited attention given to a systematic analysis of UGC in video games [9, 53, 61]. For instance, in 2017, Estrella et al. [27] linked UGC to game sales but lacked detail on in-game content like mods that directly affect rules or assets. Understanding how such UGC changes gameplay remains an underexplored but important area for future research.

As outlined in our conference paper [22], we seek to fill this academic gap by providing systematic classifications of existing UGC and related editors within the video game industry. The major contributions can be concluded as follows:

- We construct an updated Venn diagram based on the influence of UGC on video games with representative examples for each.
- We enumerate the propagation methods of UGC in video games, such as player community sharing, circulation in trading markets, and combination with original games, which boost a positive circulation of UGC and improve user's engagement.
- We propose our visions for the future development of UGC in the metaverse [23], emphasizing its role in immersive experiences, virtual economies, and social interaction within player-driven worlds.

[1]https://blog.counter-strike.net/.
[2]https://www.dota2.com/home.
[3]https://www.half-life.com/.
[4]https://playwarcraft3.com/.

However, the previous framework had notable limitations, since it does not fully capture the diversity of UGC, and the distinctions between certain categories are slightly ambiguous. Therefore, this work mainly encompasses the extension of the following aspects:

— We revise the classification using "user experience" as the key criterion, dividing UGC into two main types: changes to gameplay (expansion) and changes to resources (enhancement), with clear examples for each.
— We distinguish UGC editors from general game editors, providing representative cases to illustrate their focus on specific functionalities and integration within game environments.
— We expand the analysis across areas such as AIGC and ethical issues, providing potential research directions on the future development of UGC in video games.

The following Section 2 will offer a brief overview of the previous research. Then, we will separately elaborate the refined classification and representative cases of UGC and editors in Sections 4 and 5. In Section 7, we explore the future of video game UGC and its potential challenges, while Section 8 provides a conclusion.

2 Related Work

Research into UGC in video games has expanded significantly over the years, reflecting the growing importance of player-created content in shaping game experiences, environments, and communities. Early studies on UGC primarily focused on its role in user motivation and extending the lifecycle of video games. Lastowka et al. [53] explored the legal implications of UGC creation within virtual worlds. Daugherty et al. [17] studied the motivations driving users to create UGC. Kasapakis et al. [49] discussed best practices for leveraging UGC effectively in pervasive games using *Barbarossa* as a case. Wirman [93] identified five dimensions of player productivity in the video game industry and emphasized the importance of other users and communities to player productivity. Lindberg et al. [61] conducted a quantitative study involving surveys of members from four online video game UGC communities, which examined the influence of intrinsic motivation, extrinsic motivation, and the availability of toolkits in encouraging players to participate in the creation of UGC for video games. Researchers like Sotamaa [85] have examined the participatory nature of UGC, where players act as co-creators, enhancing the collaborative nature of game development. Moreover, some studies have examined the presentations of UGC in video games. For example, Postigo [74] analyzed user demands and developer-user conflicts through two prominent mods: *Duke It Out in Quake* for *Quake III*[5] and *GI Joe* for *Battlefield 1942*.[6] However, the existing studies mainly focus on specific game types or individual cases, while, besides our conference paper [22], there currently lacks a universal framework with comprehensive and systematic classification of UGC, which can be applied to most video games.

On the other hand, development toolkits and editors are essential for the creation of UGC, as they directly influence the user's creation experience. Several researchers have explored how these game editors affect UGC creation, leveraging various technologies. Liapis et al. [56] introduced the Sentient Sketchbook, a tool that assists designers in creating game levels by using map sketches to reduce the workload of level design. Ng et al. [68] focused on the **Augmented Reality (AR)** environment, investigating the design and concept of level editors specifically for AR games. Currently, some games have begun experimenting with new co-creation models, granting players' creative rights to unlock their potential and extend the game's lifespan. In the context of co-creation in video games, Davidovici-Nora [18] highlighted a hybrid innovation model implemented in

[5] http://www.q3arena.com/.
[6] https://en.wikipedia.org/wiki/Battlefield_1942.

World of Warcraft (WoW),[7] a popular **Massively Multiplayer Online Role-Playing Game (MMORPG)**. In addition, the integration of **Artificial Intelligence (AI)** in user content generation has gained significant attention from the academia, since AI can significantly lower the barriers to user creation and reduce production costs. For instance, Summerville et al. [86] examined the use of Long Short-Term Memory recurrent neural networks to generate levels for *Super Mario Bros.* Guzdial et al. [34] introduced an unsupervised process for generating full game levels based on gameplay video. Awiszus et al. [3] introduced "World-GAN," a generative model for *Minecraft*[8] worlds using **Natural Language Processing (NLP)** technology to create larger, more complex worlds with specified terrain. Furthermore, Guzdial et al. [33] discussed in detail how the design of AI-driven level editors influences creators' experiences. However, previous researches have primarily focused on innovations in the design and implementation of tools and editors for generating UGC, without clearly distinguishing between UGC editors and other general game editors. Besides, the editors mentioned above remain largely inaccessible to users, with their primary target users still being professional game level designers. In summary, there is a lack of systematic classification and comprehensive reviews of the existing UGC editors.

3 Methodology

3.1 Overview

This article adopts a taxonomy-driven approach, supported by a systematic review protocol to ensure both comprehensiveness and replicability. First, we construct a flexible and parallel classification framework based on users' gameplay experiences and creative intentions. Then, we perform a structured literature search, primarily using the ACM Digital Library, to map existing studies to each taxonomy branch. This method allows us to systematically organize prior work, identify research gaps, and highlight representative prototypes. The combined approach ensures both conceptual clarity and comprehensive coverage of UGC types in video games.

3.2 Construction of the Taxonomy

Our conference version has taken a decision tree-style classification, but we also realized that UGC lends itself more naturally to a parallel classification approach. According to J. R. Quinlan's research [76], dividing UGC into rigid categories is insufficient since some modifications might simultaneously change the default rules and game resources. For those which changed the resource files to create a new game mode, the original decision tree-style classification Venn diagram would have ignored changes to the game resource file, because the decision to determine whether it was a game mode change occurred in front of the resource file, and once the judgment was true, it would not be further judged. This classification treats UGC categories as mutually exclusive, when in reality, most UGC is a blend of multiple types. Given these overlaps, a more flexible classification framework is necessary. Furthermore, the original paper's four-category classification does not fully encapsulate the range of UGC in video games.

Based on the preliminary observation of the existing literature, we develop a parallel classification framework grounded in users' gaming experiences. Drawing inspiration from previous typologies such as Postigo's three-category model of modders, mappers, and skinners [73], we propose a user-centered taxonomy based on two fundamental dimensions: gameplay and resource files.

Each dimension is further subdivided into mutually non-exclusive categories. For gameplay changes, we distinguish between modifications that alter the game's core mechanics (game modes) and those that retain the base mechanics while introducing new challenges (game levels/campaigns).

[7] https://worldofwarcraft.blizzard.com.
[8] https://www.minecraft.net.

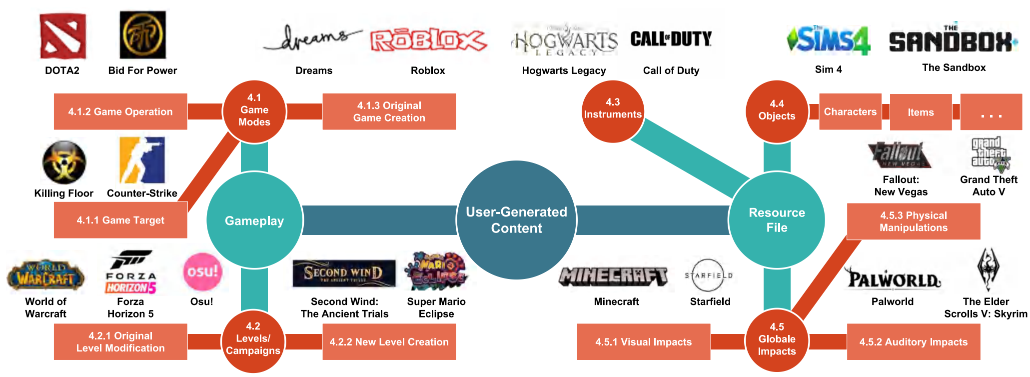


Fig. 1. Parallel classification diagram of UGC and corresponding representative games for each type.

Resource file changes are categorized into new instruments, objects, and global impacts based on their functional roles in the game environment.

The complete classification is shown in Figure 1. Our taxonomy emphasizes parallel relationships rather than hierarchical nesting, capturing the hybrid nature of most UGC. Unlike previous rigid or linear frameworks, our approach accommodates overlapping types and better reflects real-world video game instances. This taxonomy serves not only as a conceptual framework for organizing examples, but also as a structural basis for mapping and evaluating existing literature in a consistent and meaningful way. More detailed subcategories and classification rules are elaborated in Section 4. For the purpose of literature retrieval and mapping, we limit our search to the first two levels, as academic research on UGC specifically in video games remains sparse.

3.3 Systematic Literature Search

To support and validate our taxonomy-driven analysis, we conducted a structured literature search to collect scholarly works corresponding to each major category of UGC. The literature search aimed to cover diverse perspectives, including technical innovations, design considerations, user behavior, and tool development in the context of UGC.

3.3.1 Data Source. We focused on peer-reviewed publications indexed in the following major academic databases:

- *ACM Digital Library* (primary source, widely used in game research and Human Computer Interaction (HCI))
- *IEEE Xplore* (especially for technical papers on tool design)
- *Google Scholar* (for broader searches and gray literature identification)

Additionally, to supplement the academic literature, we also include grey literature such as developer blogs, community forums, and documentation from modding platforms (e.g., Steam Workshop, Nexus Mods), as these represent valuable sources of real-world UGC practices often not covered in peer-reviewed publications.

3.3.2 Search Query Design. Search queries were designed based on each taxonomy branch. Boolean logic and synonym expansion were used to capture variations in terminology. Table 1 lists sample keyword combinations used in the search.

Besides, we limited the publication date range to 2010–2025 to reflect the contemporary development of UGC within the context of modern video games. Since 2010, the proliferation of game

Table 1. Search Queries by Taxonomy Category

Taxonomy Category	Sample Search Keywords
Game Modes	"game mode" OR "mechanics" OR "gameplay"
Game Levels/Campaigns	"level" OR "campaign" OR "difficulty" OR "generation"
Game Instruments	"plug-in" OR "modification" OR "modding" OR "cheating"
Game Objects	"terrain" OR "asset" OR "character" OR "map" OR "building"
Global Impacts	"device" OR "sound effect" OR "artistic style" OR "visual"

Table 2. Inclusion and Exclusion Criteria

Criteria	Inclusion	Exclusion
Language	English-language publications	Non-English studies
Publication Type	Peer-reviewed journals or major academic conference proceedings	Abstracts, short opinion pieces, non-research articles
Research Domain	Video games and related UGC	UGC studies in unrelated domains (e.g., social media)
Content Scope	Focus on in-game UGC or game modification practices	UGC external to games (e.g., fan fictions and tutorials)
User Interaction	Tools, editors, or platforms enabling or supporting UGC creation	Procedural generation without user control or interaction

creation platforms (e.g., *Minecraft*, *Roblox*[9]) and the increasing integration of user-editing toolkits have significantly changed the landscape of UGC. Earlier literature, while relevant in historical context, tends to focus on broader media or non-interactive content and lacks alignment with the technological and participatory models seen in the past decade.

3.3.3 Inclusion and Exclusion Criteria. These criteria were designed to filter out sources that fall outside the scope of our research, such as studies unrelated to video games or those lacking academic rigor, which ensure the relevance and quality of the literature. We focused specifically on peer-reviewed English-language publications that examine in-game UGC or game modification practices. In addition, we emphasized works that involve direct user interaction with tools or platforms for content creation, rather than passive or automated content generation. The detailed inclusion and exclusion conditions are summarized in Table 2.

3.3.4 Selection Process. As shown in Figure 2, we conducted a four-stage screening following the PRISMA guidelines [65], after removing duplicates. Papers were mapped to one or more branches of the taxonomy based on their primary research focus, tools discussed, or types of UGC evaluated. Where ambiguity arose (e.g., overlap between level and mode modification), classification was determined by the authors through consensus discussion.

3.4 Literature Mapped to the Taxonomy

3.4.1 Game Modes. Existing studies emphasize player agency in reshaping game mechanics. Moody [66] offers in-depth insight into modding communities, while Comunello and Mulargia [13]

[9]https://www.roblox.com/.

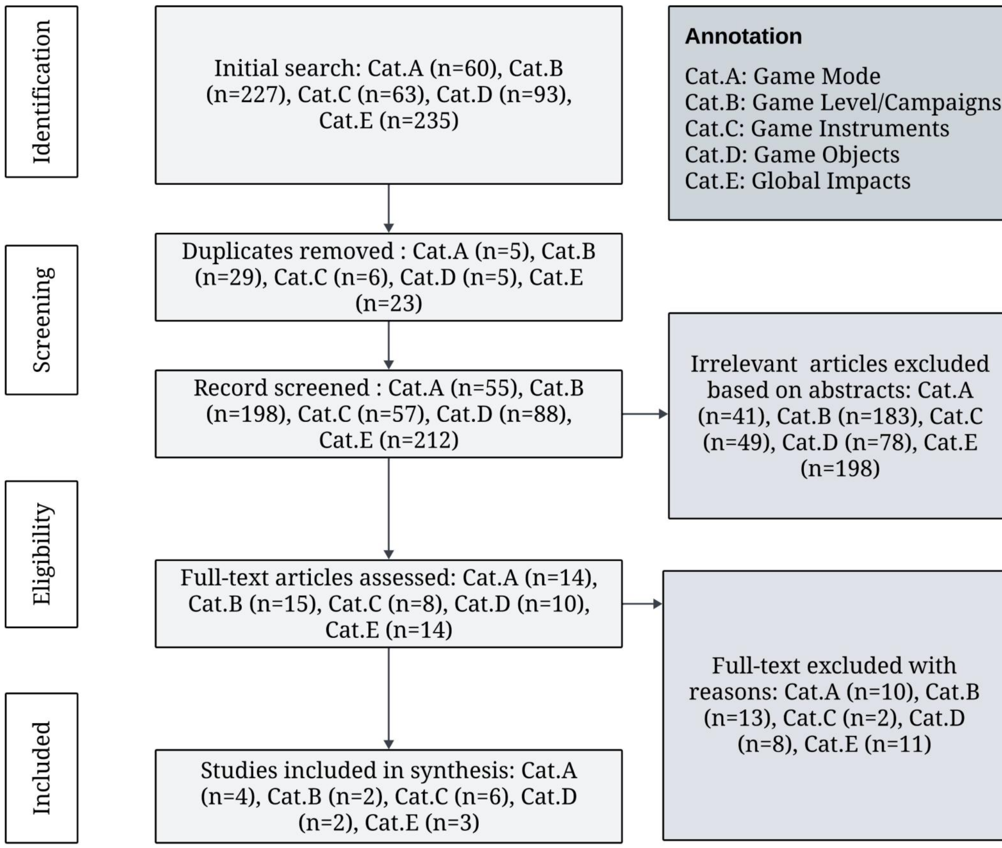


Fig. 2. A flow diagram of the article selection process.

examine *Little Big Planet* as a case of user participation in modifying the game rules. Kasapakis and Gavalas [49] provide a broad survey of pervasive games with UGC support. Kou and Gui [51] investigate user-created virtual worlds in *Roblox*, noting both creative freedom and design risks.

Although these studies differ in methodology (ranging from ethnography [66] to platform-centered analyses [13, 51] and system-level surveys [49]), they collectively affirm the conceptual shift from players as consumers to players as co-designers. However, they diverge in their assumptions about what constitutes "game mode modification": Moody [66] emphasizes emergent community practices, while Comunello and Mulargia [13] focus on in-game affordances intentionally crafted by developers, and Kou and Gui [51] foreground the governance challenges introduced by unrestricted creation. This variation reveals a lack of consensus on whether user-modified gameplay should be defined primarily in terms of technical changes, participatory design structures, or socio-cultural implications.

Across this category, the main limitations stem from the narrow scope of each study: most focus on a single platform or genre [13, 51], with limited comparative analysis across different types of game ecosystems. Moreover, few works examine long-term impacts, such as how user-modified game modes influence player retention, community norms, or commercial strategies. Key unresolved issues include the need for a unified conceptual understanding of user-driven gameplay modification and more systematic evaluation of the design, governance, and safety implications of allowing players to reshape core mechanics.

3.4.2 Game Levels/Campaigns. Among the screened literature, Hicks et al. [39, 40] investigate the quality identification and filtering of user-generated game levels, highlighting gameplay affordances and player behavior specifically in educational games. Ramos et al. [78] presents a GAN-based system that transforms user-created sketches into detailed height maps, which assists users to generate levels. Despite a relatively large number of works initially considered, most emphasize developing new algorithm in **Procedural Content Generation (PCG)**. They typically adopt a system-centered view, lacking focus on player interaction. For example, Hendrikx et al. [38] provides a comprehensive survey of PCG technique for video games which related to level generation but with minimal user involvement.

Although these studies share an interest in level construction, they differ substantially in their focal points: Hicks et al. [39, 40] prioritize player interaction and learning outcomes, Ramos et al. [78] contribute algorithmic support tools designed to augment user creativity, and PCG-oriented research [38] approaches the topic from a system-centered angle that minimizes user agency. This divergence reveals a fragmented research landscape in which user-driven level design is often overshadowed by automated generation methods. Moreover, existing studies seldom investigate how players iteratively refine or evaluate levels, nor do they examine community-level dynamics such as collaborative creation or peer review.

The main limitations in this category stem from the scarcity of user-focused empirical work and the dominance of algorithmic research that only indirectly relates to UGC. As a result, several questions remain unresolved: How do players conceptualize quality in user-created levels/campaigns? What design tools best support different levels of expertise? And to what extent do communities influence the evolution, curation, and dissemination of player-created levels/campaigns? Addressing these gaps would provide a more holistic understanding of user-generated levels/campaigns beyond algorithmic assistance alone.

3.4.3 Game Instruments. The selected studies fall into two main directions: modding communities and cheating tools. Poretski and Arazy [72] examine the social and collaborative dimensions of modding communities, while Kretzschmar and Stanfill [52] provide a typology of game mods, highlighting how mods serve as both creative tools and points of contention. Qian et al. [75] and Nhu et al. [70] investigate anomaly detection and defensive approaches against computer vision-based cheating tools in competitive games such as *DOTA 2*. Kanervisto et al. [48] develop machine learning-driven aimbots generated by GANs for cheating in First Person Shooters. Rendenbach and Dyrda [79] propose formal game space models supporting gameplay-based anti-cheating mechanisms.

Although these studies collectively explore both legitimate and illegitimate uses of game instruments, they differ markedly in their underlying assumptions. Modding-oriented research [52, 72] frames tools as enablers of creativity and participatory culture, emphasizing social organization, identity, and governance within creator communities. Cheating-related research [48, 70, 75, 79], however, adopts a technical and adversarial stance, treating instruments as vectors of exploitation that threaten fairness and competitive integrity. This divergence creates a conceptual gap: while one stream focuses on the positive role of tools in expanding gameplay possibilities, the other focuses on stopping harmful tool usage, with little work examining how legitimate modification tools and illicit cheating mechanisms may share underlying technological foundations or influence each other.

A key limitation of the existing research is that legitimate game instruments, such as enhancement tools, accessibility utilities, or user-created automation scripts, receive comparatively little empirical or conceptual attention. Most studies gravitate toward either community-level modding dynamics or high-stakes cheating scenarios, leaving the middle ground largely unexplored. Unresolved questions remain regarding how legitimate instrument-based modifications shape player experience, how developers might better integrate user-created tools into game ecosystems, and how boundaries between acceptable and unacceptable modifications are negotiated across different communities and game genres. Considering these questions would deepen understanding of game instruments as a multifaceted component of UGC ecosystems.

3.4.4 Game Objects. Current research under this category primarily centers on the customization of characters and terrains modifications within video games. Zomerplaag and Bakkes [100] explore identity expression through character customization, while Low et al. [59] examine cooperative placemaking in sandbox games as a form of UGC.

In contrast, a substantial body of work explores automated terrain and environment generation, often with minimal involvement from players. For example, Lindley et al. [58], Togelius et al. [29], and Broadbent et al. [8] propose various generative techniques (ranging from evolutionary computation to data-driven synthesis) for producing landscapes, structures, or environmental features. However, these PCG-oriented approaches typically adopt a system-centered view, prioritizing algorithmic efficiency, realism, or scalability over user interaction. As a result, the connection between procedural terrain generation and player-driven creative practices remains weakly explored.

Although both streams address object-level modification, they operate with very different assumptions: customization research [59, 100] emphasizes meaning-making, social collaboration, and representational agency, whereas generative terrain research [8, 29, 58] frames objects as technical products of algorithms rather than user creation. This divergence highlights a conceptual gap regarding how object-level modifications mediate the relationship between players and game worlds. For instance, little is known about how users incorporate procedurally generated terrains into creative workflows, how object-level tools influence player motivation, or how collaborative object creation shapes community structures. Current limitations include the scarcity of empirical studies examining how players actually engage with terrain-editing tools or object customization interfaces, and the lack of research bridging automated generation with user-centered modification practices. Unresolved questions include the need to understand the co-evolution of player creativity and PCG systems, the social implications of shared object creation, and the ways in which object-level tools can better support expressive or inclusive play. Addressing these gaps would deepen insight into game objects as a core component of UGC ecosystems rather than solely algorithmic artifacts.

3.4.5 Global Impacts. Existing studies predominantly investigate how in-game audio, visual elements, and input devices affect user experience and biofeedback. Methodologies range from experimental evaluations of sound design and subliminal visual cues to analyses of controller types and haptic feedback systems. Thiparpaku et al. [91] compare sword sound effects across two games to assess their impact on immersion and satisfaction, while Maureira et al. [62] and Caroux [11] focus on user responses to visual features. Hufnal et al. [42] examine the influence of different controller types (such as gamepads and mice) on the user experience of VR games. Zhang et al. [98] introduce a glove sleeve with haptic feedback to enhance the physical interaction in flight video games, which can be considered as a change of physical input.

Although these studies share an interest in the sensory and physical dimensions of gameplay, they adopt diverse methodological approaches: controlled experiments on audio [91], visual-cue analysis [11, 62], device-comparison studies [42], and prototype-based evaluations of haptics [98]. However, they converge on a system-centered perspective in which audio, visual, and input features are designed and manipulated exclusively by developers or researchers. This creates a notable contrast with UGC-oriented categories, where players actively shape game content. In the global-impact literature, users are primarily treated as evaluators rather than creators, and the potential for player-driven modifications of soundscapes, visual environments, or control schemas is largely overlooked.

The main limitation in this category is the absence of empirical work examining how players themselves alter global-impact elements, such as customizing sound packs, reshaping lighting or visual filters, or creating new control mappings. These practices are increasingly common in contemporary game communities but remain underexplored in academic research. Unresolved questions include how user-modified global elements influence immersion, accessibility, or competitive balance; how such modifications spread within communities; and how developers might integrate user-driven global-impact tools without compromising design coherence or fairness. Studying these points would expand understanding of global impacts from purely developer-defined features toward a more participatory, user-influenced model of audiovisual and interaction design.

4 Categories of UGC

As Section 3 mentioned, we first divide UGC into two primary categories:

(1) *The change of gameplay*: By modifying the default rules or adding new levels, users can expand their gaming experience. Adams [2] provided a non-rigorous definition of gameplay, dividing it into two aspects: the "challenge" presented to players and the "actions" they need to take. Building on this work, we broaden the definitions of "challenge" and "action" to encompass all possible gameplay situations. Generally speaking, we classify the change of gameplay into two directions including game modes and game levels/campaigns. The game mode changes are more focused on the core rules and mechanics of the game, while the changes of game levels/campaigns tend to rely on existing modes to generate new challenges. Note that, according to the study of Adams [2], we consider that the emphasis of the change of gameplay should focus on the novelty of experience comparing to the change of resource file. For example, uploading a user-made map without any change of game framework or storylines should be considered as a resource file.

(2) *The change of resource files*: Users can modify the game files or upload new resources to enhance their gaming experience. The most common mods in the video game community typically involve resource files that do not alter gameplay or core mechanics. Instead, they achieve various purposes by adding or overwriting files in the game's installation folder. These modifications can range from practical improvements, like automating repetitive tasks, to more playful changes, such as replacing character models with real-life celebrities in RPGs. In this article, we consider to classify these resource files into three categories: new objects, new tools, and global modifications.

We construct a parallel classification bubble diagram with corresponding representative games for each type (Figure 1). Note that the UGC we study is categorized from the perspective of the user, which does not mean that each game belongs to only one specific category, in other words, users may create different contents within the same game. Moreover, since there are many types of game objects, only a few representative examples are selected for analysis in this article (e.g., *Sims 4* and *The Sandbox*). In the following subsections, we will clarify the definition and further discuss the characteristics and prototypes in detail.

4.1 User-Generated Game Modes

User-generated game modes introduce innovative gameplay experiences by modifying fundamental rules or objectives. In this article, we assert that the emphasis on game modes should reflect the mechanics of the original games. As mentioned in the conference version [22], new game modes need not be entirely distinct from their base games; rather, they should provide novel gaming experiences that deviate from the original rules or targets. To better understand how these modifications manifest across different game types, we categorize them into three aspects: game targets, operations, and original game creation. Each of these categories will be elaborated upon in the following sections.

(a) *Half-Life* (b) *Counter-Strike* (c) *Unreal Tournament* (d) *Killing Floor*

Fig. 3. Game target changes in FPS games.

4.1.1 The Change of Game Target. The target of the game often directly determines the condition to win in most competitive games, i.e., a new game mode can be generated by changing some rules of the original game. The most notable case is the aforementioned *CS* (Figure 3(b)), which is originally developed by Minh "Gooseman" Le and Jess Cliffe in 1999. As a modification of *Half-Life* (Figure 3(a)), it changed the gameplay of *Half-Life*, which required users to solve puzzles and follow the storyline, instead it divided players into two teams, terrorists and the counter-terrorists, to fight against each other. Depending on the map, there are four types of scenarios:[10] hostage rescue, bomb defusal, assassination and escape, where the official maps of the fourth mission type "escape" has been removed before the game's release. The creators of the initial modification have obviously changed the game target of *Half-Life*, abandoning chapters that rely on story support. Compared with the team deathmatch mode that players are fighting alone most of the time, *CS* chooses to rely on the direct confrontation between two opposing teams to win.

The change of game target is usually more intuitive and less connected to the original work. Therefore, we believe that changing the game target means changing some specific rules of the basic game, or even the storyline. In other word, it is more appropriate to be regarded as a standalone spin-off game. For another example, *Killing Floor*[11] (Figure 3(d)) was released in 2005 as a mod for *Unreal Tournament 2004 (UT)*[12] (Figure 3(c)) which had completely different game modes. *UT* is an arena shooter focused on fast-paced, competitive multiplayer matches. The game target depends on different game mode but usually revolves around killing opponents to score points or capturing objectives to win the match. But *Killing Floor* aims to survive multiple waves of horror-themed enemies and defeat the final boss in a cooperative setting. In *Killing Floor*, the creators also changed the sci-fi theme from the original to a gruesome and bloody theme filled with zombie-like creatures, which could also be regarded as a change in the game theme, but here we consider the change of game target is dominant.

4.1.2 The Change of Game Operation. Game operation refers to the mechanics of how the game is played, distinct from the game's target or objective. While a modification changes the game's target, the core gameplay mechanics, or operations, might remain consistent. For instance, *CS* mentioned above has changed the rules of winning the game, but in terms of operation, it is still a FPS game. However, changing the game operation often involves changing the game target. A simple case is altering the game's perspective, such as switching from first-person to third-person view, without changing other gameplay elements. If more modifications are added, the target and tempo of the game may also change. *Bid For Power*[13] is a complete conversion for *Quake III* with gameplay that is quite different from the original game. It changes the way the original first-person shooter works, allowing players to control the Ki-powered superhero from a third-person perspective, and it is faster paced than the original game.

[10] https://counterstrike.fandom.com/wiki/Map.
[11] https://www.killingfloor2.com/.
[12] https://www.utzone.de/.
[13] https://bidforpower.net/.

(a) *Warcraft III* (b) *Defense of the Ancients*

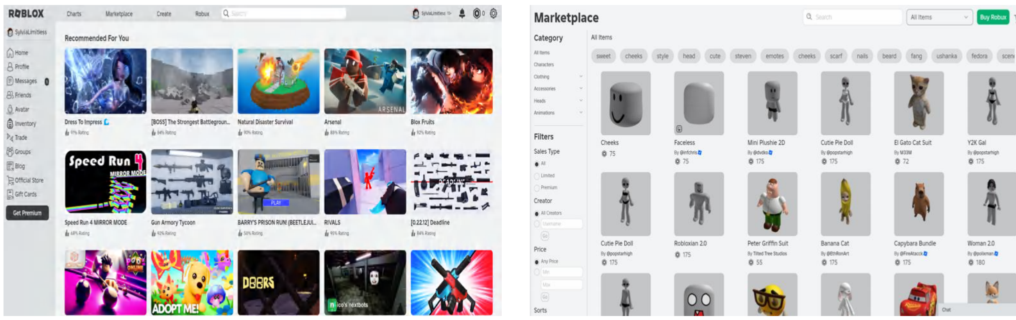
Fig. 4. Game operation changes in *DOTA*.

In the context of more complex modifications, we argue that the concepts discussed in this article can be integrated. User-generated modifications can alter both game operations and targets, resulting in a comprehensive shift in gameplay dynamics. A representative case is *DOTA* (Figure 4(b)). Originally, the *Warcraft III: Reign of Chaos* (Figure 4(a)) allows players to command armies to fight against enemies' bases in a **Real-Time Strategy (RTS)** game, while *DOTA* changed the mode of controlling an army to only controlling a hero character. Comparing with *Warcraft III*, *DOTA* divides players into two teams of five for 45–90 minutes, with each player choosing the unique hero to fight. From the gameplay point of view, *Warcraft III* is more open, and this feature is more prominent in the subsequent *WOW*. Although *DOTA* and *Warcraft III* share similarities in worldview and character setting, as Remi Smith [84] says, just like chess and Monopoly that both require strategy and experience, but they are fundamentally different.

4.1.3 Original Game Creation. This category differentiates itself from traditional game modifications, as illustrated by *Roblox*. These games only provide basic rules, environments, materials, and operations, and players can create entirely new game modes by altering both gameplay objectives and mechanics. These alterations include changes to both game targets and operations, while remaining closely aligned with the original framework. In this way, the original game creation transforms the original game into an open platform, functioning like a game editor or a resource library for users. *Roblox* encourages users to generate their own game using internal resources and upload it on the community (Figure 5(a)) so that other users can easily acquire what they are interested in and even pay for it through the marketplace (Figure 5(b)). While some developers offer corresponding game editors, few platforms provide the same level of freedom as *Roblox*. The use of *Roblox Studio* allows players to create games with greater ease, effectively reducing barriers associated with modifying game code. As noted by Young joo et al. [46], *Roblox* blurs the distinction between game creators and players. Comparing with other games, this platform functions similarly to a marketplace, where players can buy and sell content, while developers are responsible for establishing rules and maintaining order within the ecosystem.

"*Dreams*",[14] developed by *Media Molecule*[15] for the *Play Station 4*,[16] serves as another notable example of original game creations. As an innovative and expansive game, it offers robust development tools that give players a high degree of freedom to create and share various types of content, including animations, games, and music. Users can design everything from simple mini-games to complex, fully-featured experiences, utilizing the platform's capabilities to sculpt 3D models, apply

[14] https://www.playstation.com/de-de/games/dreams/.
[15] https://www.mediamolecule.com/.
[16] https://www.playstation.com/de-de/ps4/.



(a) Game Community (b) Marketplace

Fig. 5. *Roblox* encourages multiple original game creations.

textures, create animations, compose music, and develop interactive elements through an intuitive interface. Both *Roblox* and "*Dreams*" illustrate the potential of UGC to transform traditional gaming experiences into open, collaborative platforms.

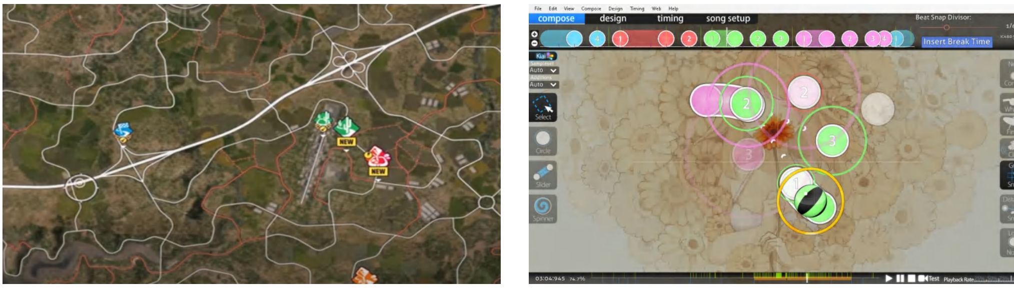
4.2 User-Generated Game Levels

As another type of gameplay change, user-generated levels can be regarded as a continuation of the original game. Instead of introducing new rules, operations or win conditions through code or external tools, level modifications retain the original rules but offer new content within those constraints. Such changes do not usually result in an independent game, as they depend on the base game's mechanics, assets, and structure. Users who, after completing the game, may seek additional challenges or wish to express their creativity by generating new levels. We categorize user-generated game levels into two main types based on their degree of deviation from the original design: original level modification and new level creation. The following subsections will analyze their characteristics and discuss some typical examples, respectively.

4.2.1 Original Level Modification. These are modified versions of the original game levels where the structure remains recognizable, but certain elements such as enemy counts or time limit are adjusted to change the difficulty or path. The key criterion is whether the modified level retains certain similarities to the original in aspects like terrain or storyline.

The simplest kind of modification based on the original level is to change the difficulty. These changes do not alter the narrative or core challenges but instead adjust in-game variables like enemy health and damage. For example, in MMORPGs like *WOW*, factors such as cooldown times or skill damage affect the **Time-to-Kill (TTK)**. In card games, deck size or turn time influence player pressure. In adventure or roguelike games, changing the protagonist's health or reducing the number of revivals is the quickest way to change game difficulty. Therefore, game developers usually set different levels of difficulty for the same level in order to stimulate players' interest. It is important to distinguish level modifications from other category "game instruments" such as unlimited ammunition. The difference lies in scope: difficulty changes to levels must have an impact on the overall situation rather than individuals. For example, in multiplayer games, merely changing the attack damage of a single player does not change the difficulty of the entire level for other players.

Another modification is focus on the game path. Here, we define the word "path" as the approach or route to achieve the game target. A game may offer multiple paths to win, e.g., in the bomb defusal mode of *CS*, terrorist can win the game either by eliminating all enemies or by planting a C4



(a) EventLab in *Forza Horizon 5* (b) Beatmap editor in *Osu!*

Fig. 6. New game path creation.

bomb in the limited time. In this case, we consider that "kill all enemies" and "let the bomb explode" as two different paths. However, this is only an explanation to better understand the definition of game path, while the change of game path we study in this article does not involve the change of game operation. Racing games offer intuitive examples. In *Forza Horizon 5*,[17] players can design custom routes to the same destination using *EventLab* by placing checkpoints and obstacles in the open world, as shown in Figure 6(a). Another example is rhythm game, in which there are different beatmaps for the same song depending on the difficulty. As shown in Figure 6(b), *Osu!*[18] supports users to make new beatmaps for specific songs using its built-in editor. In terms of the game rules, changing beatmaps does not change the core mechanics, while players essentially still need to score points by hitting notes in circles.

4.2.2 New Level Creation. These are entirely new levels designed by users which are different from the existing levels except for the basic mechanisms and rules, such as having new environments, tasks or storylines. Compared with modifying the original levels, designing new levels can break free from the limitations of the original framework and give greater play to the creativity of users.

In platformers like *Super Mario Sunshine* (Figure 7(a)), for example, the fan-made mod *Super Mario Eclipse*[19] introduces a user-generated game world within many new levels and challenges. Generally, the form of a new level varies across different types of games. In open-world role-playing games with high degree of freedom, it often takes the shape of additional stories, such as interacting with special characters to trigger separate side stories. *Second Wind: The Ancient Trials*[20] (Figure 7(b)), a popular unofficial expansion for *The Legend of Zelda: Breath of the Wild*,[21] demonstrates this clearly. Developed by the community for players who had already completed the base game, it adds new locations, side quests, and characters while preserving the underlying systems of the game, such as combat, traversal, and puzzle solving. These additions provide players with new content and goals, deeply enriching the gameplay without altering its core logic.

These types of creations present how users can leverage a game's original system to deliver what feels like an entirely new experience. Unlike game mode modifications, new level creation does not involve changes at the system level, such as altering user interaction methods or introducing new mechanisms. As the "challenges" presented to players according to Adams [2], it should not alter

[17]https://forza.net/horizon.
[18]https://osu.ppy.sh/.
[19]https://gamebanana.com/mods/432762.
[20]https://gamebanana.com/projects/35468.
[21]https://zelda.nintendo.com/breath-of-the-wild/.

(a) *Super Mario Eclipse* (b) *Second Wind: The Ancient Trials*

Fig. 7. New level creation examples.

the "actions" that players have to take when facing the challenges. Ultimately, new level creation allows players to escape the spatial, narrative, and design constraints of official levels, granting them creative freedom to construct entirely new experiences built upon familiar rules. Although these creations may resemble standalone games in scope or presentation, they remain inherently dependent on the original framework. Without the foundational support of the original game, these expansions cannot function on their own. From a purpose-driven perspective, players create expansion packs not to develop standalone games but to extend their experience within a game they have likely spent hundreds of hours playing.

4.3 User-Generated Instruments

User-generated instrument is a common form of modification, which refers to modifications that alter specific components or features of a game—typically with the goal of optimizing users' experience without changing the game's fundamental rules or core systems. These modifications can be used to either skip repetitive or monotonous levels, or to adjust the game's complexity. In many single-player games, user-generated instruments developed by individual players or third-party creators are widely adopted due to their small scale and ease of integration. For example, in *Hogwarts Legacy*,[22] using a spell to unlock certain rooms still requires solving simple puzzles, a process that can become tedious with repetition. In this case, some players have made modifications to skip the unlocking mechanics, which greatly saves time. Within the player community of *Stardew Valley*,[23] numerous Mods exist that can change the character profiles and dialogue content. The widespread adoption of these tools has greatly expanded the possibilities for customization in gaming, which allows players to obtain unique gaming experiences by choose different mods on the same game. Although most instruments exist in the form of mods, our classification of UGC categories in this article is not solely based on their existence forms. As previously mentioned, *CS* originally emerged as a mod for *Half-Life*, yet it eventually evolved into a standalone title. This illustrates that the nature of UGC should be understood in terms of its functional role and impact on gameplay, rather than its form of implementation alone.

However, many user-generated instruments have the potential to disrupt the game balance in order to improve individual game experience. These tools, often referred to as "plugins," are frequently used as cheats, particularly in multiplayer environments. Popular modifications like "Auto Target Enemies" reduce TTK, creating an unfair advantage over other players. In some competitive games, such as *Call of Duty*'s[24] multiplayer mode, the use of cheat "plugins" is very

[22] https://www.hogwartslegacy.com.
[23] https://www.stardewvalley.net/.
[24] https://www.callofduty.com.

Table 3. Well-Known Cheating Codes in *Call of Duty*

Cheating Code	Effect
Kill	Kills the player.
Notarget	Lets enemies to ignore you.
Give Ammo	Replenish the current weapons and ammunition.
Give All	Obtain all weapons and full ammunition.
God	Player won't die.

common. Table 3 shows some well-known cheating codes which modify the properties of program files to attain desired effects.[25] Game developers often crack down on the use of such cheating tools, but we consider that these "plugins" are indeed user-generated instruments. The abuse of these "plugins" involves ethical problems, where the relationship between the existence of plugin cheats and game fairness will be further discussed in detail in Section 7.2.

In some cases, excessive use of user-generated instruments—i.e., an overabundance of modifications—can lead to qualitative changes that alter the game's core mechanics. For instance, in a standard competitive shooting mode where players are allowed to respawn at certain points after death, adding a "no resurrection" modification along with an increase in weapon damage could transform the normal mode into an only-one-life mode, resulting in a more intense game tempo. It is important to note that, in this article, separating the instruments and game modes does not mean that the two are completely opposed.

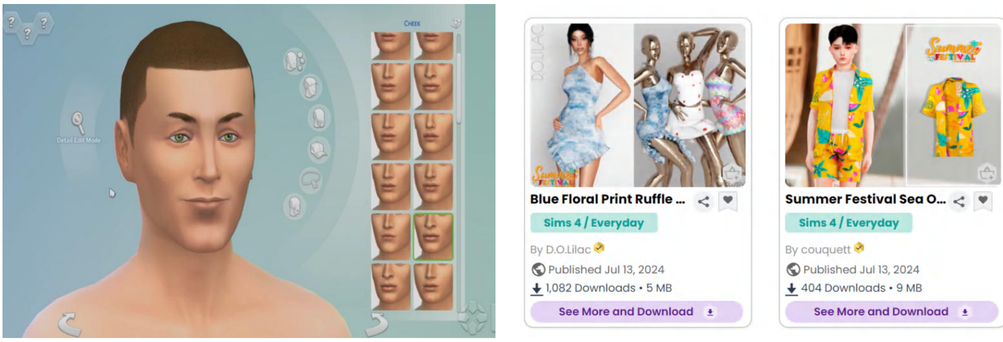
4.4 User-Generated Objects

The definition of objects is very broad, where we consider that game characters, items, maps, and so on, as resource files of the game, can be regarded as game objects. In most game mods, altering or adding game objects is one of the most common forms of modification. Therein, one of the most typical changes of game objects is the personalization of characters. Many role-playing games allow players to customize their character's appearance, including aspects like body shape, skin tone, and facial features, offering a tailored gaming experience that reflects individual preferences.

The game *Sims 4*[26] leverages this characteristic as a major selling point, giving players the freedom to create unique characters and manipulate them to follow the personalized storylines. Even though the developers consistently release expansion packs with new themes such as seasons and high school years, many players are still not satisfied with the existing resources provided by the official developers. Mods address this gap by providing a vast array of additional content created by the community. Notably, platforms like *Nexus mods* offer a variety of complete modifications for players to download. As shown in Figure 8(b), new clothing will be added by the corresponding mod, besides there also exist many modifications of houses, furniture, and other objects. Like the unique characters generated by the player through the built-in editor (Figure 8(a)), these external resources introduced by mods are also parts of user-generated objects.

In addition to common objects, there is a special case—maps. Maps are an indispensable part of most video games and play a great role. Here, we use *Minecraft* as a prototype to make a distinction: If a map serves only as a visual or architectural showcase without embedded game rules—such as a structure designed purely for viewing—it should be categorized as a resource file. However, if a map incorporates interactive elements, such as a secret escape room or a detailed storyline

[25] https://gamefaqs.gamespot.com/pc/914586-call-of-duty/cheats.
[26] https://www.ea.com/games/the-sims/the-sims-4.



(a) Built-in editor in *Sims 4*

(b) External game objects available in game community

Fig. 8. User-generated game objects in *Sims 4*.

and background, it should be considered as a gameplay modification. For example, the famous modification "*Twilight Forest*",[27] which introduces a forest world shrouded in eternal night allows players to beat several powerful bosses as adventurers to earn rewards. While game objects are often integral to gameplay modifications, they are fundamentally types of resource files. Their significance primarily depends on their impact on gameplay, as not all user-generated object changes directly alter gameplay dynamics.

4.5 User-Generated Global Impacts

To begin, we must clarify the definition of user-generated global impacts, as it can often be confused with changes to game themes or user-generated maps. A global impact can be defined as a modification that alters the player's sensory experience—specifically visual, auditory, or tactile sensations—without affecting gameplay mechanics or the original resource files. In essence, global impacts are modifications that adjust how the game feels rather than how it functions. Generally, we conclude three main sensory involvements of players, including visual impacts, auditory impacts, and physical manipulation. In the following sections, we will examine these global impacts individually, using representative examples to illustrate each type.

4.5.1 Visual Impacts. Visual impact changes are among the most intuitive user-generated modifications, typically altering the appearance of resource files. These changes range from simple color adjustments to complex texture overhauls. The most typical example is *Minecraft*, user-generated texture packs and shader packs contain textures and light change, improving the visual quality of game objects. Figure 9 shows a famous shader pack named "*Vanilla Plus Shader*"[28] which was developed by "RRe36." This shader pack enhances lighting, shadow, water effects, plant dynamics, and block materials, all while preserving the original aesthetic of vanilla *Minecraft*. In games that prioritize immersive visuals less, filter modifications are more prevalent. For instance, mods that remove the default dark green filter in *Starfield*[29] clarify the visual display, improving overall screen visibility. Additionally, certain filters can enhance the gaming experience for color-blind players, allowing for greater visual accessibility.

[27]https://github.com/TeamTwilight/twilightforest.
[28]https://rre36.com/vanilla-plus.
[29]https://bethesda.net/en/game/starfield.

Fig. 9. Visual Effect of *Vanilla Plus Shader*[30].

In 2022, Lin et al. [57] proposed using "impact feel" to describe how players feel when they receive feedback from juicy impact in a game. We consider that such feedback is a combination of visual and auditory effects. In their study, they trained a NLP model to rank the impact feel of action games, finding that hit stop, sound coherence, and camera control may significantly influence the impact feel. For action game players, enhancing attack effects—such as increasing color contrast, adjusting view angles, and intensifying visual details—falls within the scope of global visual impacts. Although few current UGC projects focus on enhancing impact feel, its importance in player experience suggests it may become a valuable area in future game development.

4.5.2 Auditory Impacts. We consider that background music, character voices, and all sound effects should be auditory impacts. Some games allow players to upload music files to play (while there might exist copyright issues in practice), or even replace the original sound effects with another game's sounds. A typical examples can be found in *Palworld*.[30] The mod "*Ninja Slayer Audio*"[31] created by *sg12041* substitutes the voices of player and human enemy with dialogue from the *Ninja Slayer* animation. Sometimes a game designer's intention for a particular sound is at odds with the player's experience. Many players complained that the sounds made by Lucky Pals were annoying when playing the game, so a modification called "*Shiny Lucky Pals No Sound Shining*"[32] was created to eliminate the sounds made by these creatures. The changes in the game experience brought about by changing the audio are also intuitive, and sometimes even affect the player's manipulation, such as using clearer ballistic sounds to allow the player to judge the position faster.

4.5.3 Physical Manipulations. Physical manipulation refers to the methods players use to interact with games, typically through keyboard and mouse input. However, some games—especially those from developers like *Nintendo*[33] and *PlayStation*[34]—may be limited to specific hardware or lack support for alternative controllers. To enable diverse gameplay experiences, modifications can provide users with different control options. For games such as *Fallout: New Vegas*[35] that do not support controller input, mods can be used to enable gamepad functionality, offering a more comfortable experience for some players. Additionally, specialized input methods, such as using a simulated steering wheel in *Grand Theft Auto V*,[36] allow for a driving experience closer to real

[30] https://www.pocketpair.jp/palworld.
[31] https://www.nexusmods.com/palworld/mods/771.
[32] https://www.nexusmods.com/palworld/mods/124.
[33] https://www.nintendo.com.
[34] https://www.playstation.com.
[35] https://fallout.bethesda.net/en/games/fallout-new-vegas.
[36] https://www.rockstargames.com/gta-v.

life, enhancing immersion beyond traditional keyboard and mouse controls. These modifications significantly enrich the gameplay experience by providing alternative ways to physically engage with the game world.

4.6 Lessons Learned

From the reviewed studies and industrial practice in Section 4, several overarching insights emerge: (1) Although the forms of UGC vary widely across video games, a common pattern is that users tend to create content that aligns with the affordances and constraints of the original game design, indicating that developer-provided tools strongly shape the direction of creativity. (2) UGC creation is consistently influenced by the balance between technical complexity and creative freedom: editors or mechanisms that lower the technical threshold attract broader participation, while more sophisticated tools tend to cultivate smaller but highly dedicated creator communities. (3) Existing research predominantly focuses on the functional roles of UGC, such as modifying gameplay or enhancing resources, while paying comparatively less attention to user motivation, community dynamics, and the long-term evolution of creator ecosystems. (5) We observe that UGC often fills design gaps left by game developers, extending replayability and enriching player experience in ways that official content alone may not achieve. (6) The literature highlights several underexplored areas, including the interplay between UGC and player identity, the lifecycle of UGC communities, and cross-game or cross-platform propagation of user creations. These observations collectively suggest that future research should move beyond categorization toward deeper investigation of user behavior, social structures, and design implications surrounding UGC in video games.

5 UGC Editor

A UGC editor is a tool designed for ordinary users to create in-game content without advanced programming skills [68]. Unlike professional game engines, these editors are designed as extensions of games and focus on enhancing user experience by enabling content creation, sharing, and replayability. In our conference version, we divided UGC editors into four categories: level editors, gameplay editors, game editing engines, and community-made game editors. However, the criteria were inconsistent. For example, "community-made game editors" were by definition only intended to distinguish them from official game editors, but they were not well differentiated from other genres in the UGC they created. The classification also ignored common tools such as resource editors found in many RPGs (e.g., character customization), which are widely used but unaccounted for.

To address these issues, we now classify editors based on the type of UGC they generate, regardless of whether they are official or not. As shown in the Figure 10, we build a parallel classification bubble diagram for UGC editors, divided into two categories: gameplay editors and resource editors. Each type will be discussed in the following subsections.

Additionally, we introduce the concept of "general editor" to provide a dialectical comparison with UGC editors. While common game development tools like *Unity*[37] and *Blender*[38] have been increasingly used to create UGC in recent years, they do not fit neatly into any specific category based on the classification criteria for UGC editors outlined in this article. Despite this, the content produced by users through these general editors can be categorized as UGC, and thus, the corresponding tools may be considered UGC editors under this broader classification.

5.1 Gameplay Editors

As we mentioned in Section 4, for UGC, changing the default rules or adding new levels is a change in gameplay. Based on this definition, we name the tools that can provide such UGC as the gameplay

[37] https://unity.com.
[38] https://www.blender.org/.

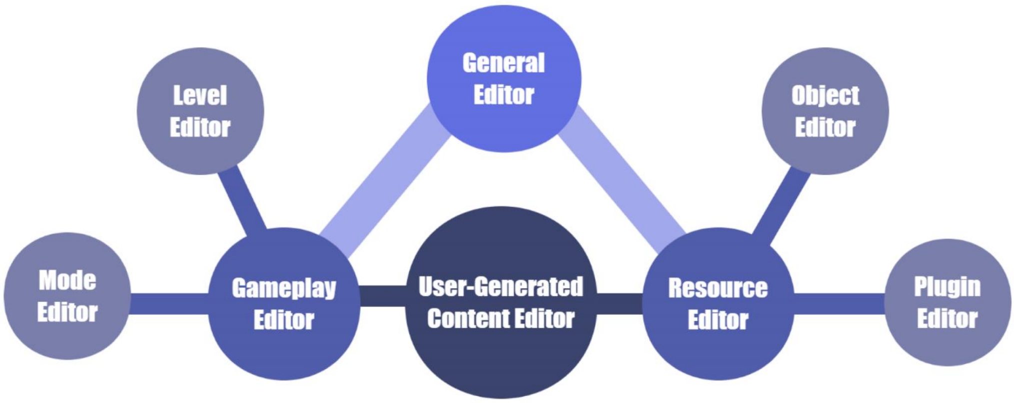


Fig. 10. Parallel classification diagram of UGC editors.

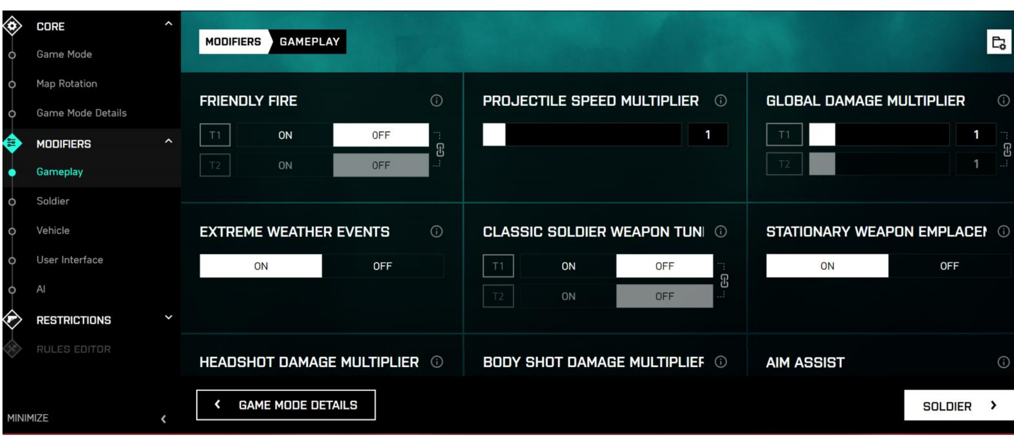


Fig. 11. The official mode editor gives players a high degree of freedom.

editors. Then, we continue to subdivide them into mode editors and level editors according to the different UGC created by these editors.

5.1.1 Mode Editors. Some **AAA (3A)** games offer customizable mode editors that allow players to modify the game mode and basic rules, such as the number of people in each match and the limited time. A typical example is *Battlefield Portal*[39] released by *Battlefield 2042*,[40] which allows players to create new game modes. Figure 11 shows the editor interface on the official website of *Battlefield 2042*. This editor clearly empowers the player to establish the game's rules. From selecting the mode and choosing the map to defining specific gameplay rules and adjusting character and weapon modifiers, the player enjoys significant freedom in customizing the experience. This is the core function of what we define as a mode editor.

Another representative example is *Warcraft III: Frozen Throne* which includes the *World Editor*, an official level editor released by *Blizzard Entertainment*.[41] The *World Editor* allows users to design

[39] https://www.ea.com/games/battlefield/battlefield-2042/game-overview/modes/battlefield-portal.
[40] https://www.ea.com/games/battlefield/battlefield-2042.
[41] https://www.blizzard.com/en-us/.

custom maps, create new units, set up triggers, and even script custom game mechanics. This powerful tool enables players, even without programming expertise, to easily create custom game modes. Some of the most popular custom games, like *DOTA* and *The Lord of the Rings*, originated from the *Warcraft III World Editor*. Although we classified the *World Editor* as a level editor in our conference version [22], it was intended to encourage players to design new maps and generate new campaigns. Therefore, we believe that editors giving players the power to reset the rules of the game should be identified as a mode editor, since it is our fundamental criterion to distinguish mode editors from level editors.

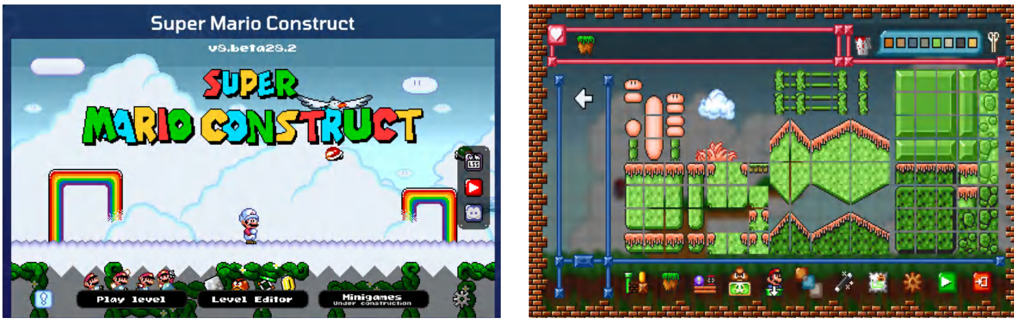
In general, creating new game modes is often more challenging than designing new levels within existing rules, as developers must carefully consider the balance of the scripts running in the game and ensure the editor's functions are both robust and precise. To minimize the likelihood of bugs, the significant potential of game mode editors comes with equally high development costs, largely due to the complexity involved in creating these advanced tools. This also explains why gameplay editors are more commonly found in 3A titles rather than indie games. Furthermore, modifying a game mode often involves altering the core mechanisms of the game, which typically requires writing game scripts and modifying the code. Unless there exist an official editor provided by the developers, it is challenging for players without a programming background to create new game modes. This is one reason why editors that allow for game modes modification are relatively rare. Additionally, creating and releasing such editors as unofficial tools can raise issues related to copyright and conflicts of interest. For example, as an open source third-party game editor, *HiveWE*[42] is not officially supported by *Warcraft III* and cannot distribute the editor for legal reasons. We will further elaborate in Section 7.2.2 on whether authorities should provide adequate support to third parties and pursue cooperative interests.

5.1.2 Level Editors. Level generation, a key challenge in PCG, has emerged as a significant research topic in recent years [50]. The concept of level editor has been well-established in recent years, with many adventure and RTS games offering tools for level design. Ipacs [44] suggests that levels are not merely a series of obstacles but rather thoughtful combinations of elements designed to provide players with new experiences, but she ignores the contradiction of whether the level should obey the rules of the game. In the context of UGC, it is significant to differentiate between levels and modes. Some editors, marketed as "level editors" (such as the *World Editor* mentioned in Section 5.1.1), may not strictly fit this classification, as they offer users extensive freedom, enabling them to produce content that may deviate from the core mechanics of the original game.

We define a level editor as a game editor that enables users to use existing elements to create new levels or campaigns using existing in-game elements without changing the game's rules or targets. Unlike mode editors, which modify the core structure of the game, level editors maintain the fundamental framework, making them more affordable to develop and widely accessible. For example, as we mentioned in Section 4.2.1, using *EventLab* to add obstacles and design a new track in *Forza Horizon 5* could be considered a new user-generated level. *Doom*[43] supports various level editors, and the latest *Ultimate Doom Builder*[44] includes multiple new features for players to edit levels in addition to the previous *GZDoom Builder*[45] features.

Creating new levels can revitalize an existing game and significantly extend its lifespan by offering users creative freedom. Communities built around these editors provide a platform for

[42] https://www.hiveworkshop.com/threads/hivewe-world-editor-0-6.303110/.
[43] https://bethesda.net/en/game/doom.
[44] https://ultimatedoombuilder.github.io/.
[45] https://github.com/m-x-d/GZDoom-Builder.



(a) Editor interface (b) Level preview

Fig. 12. Level editor in *Super Mario Construct*.

sharing and collaboration. For example, *Level Share Square*[46] is a level design community unaffiliated with any specific Intellectual Property that currently supports games from the *Super Mario series*. Figure 12 shows an online level editor called *Super Mario Construct*,[47] with a concise editing page. In general, a level editor typically provides users with predefined in-game elements, limiting the extent of customization. For instance, users can modify aspects like the color scheme of obstacles, but they generally cannot change models by importing external resources. On the other hand, the combination of browser-based editor and community is not uncommon, some 2D adventure games seem to be more suitable for such type of level editor, which does not require to download additional software, reducing the development cost.

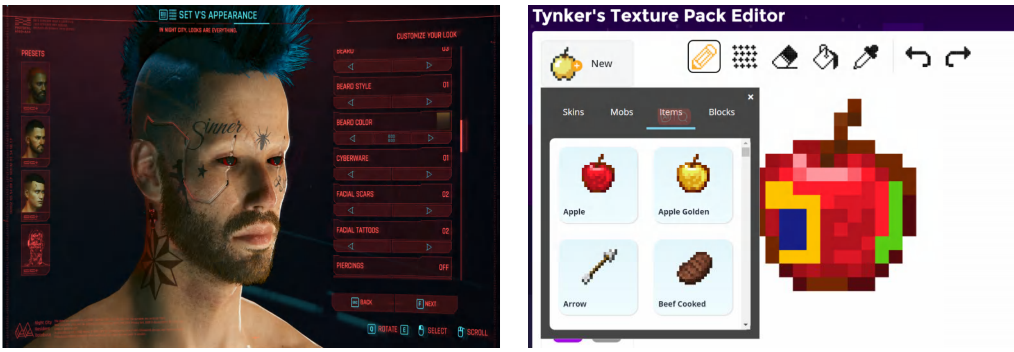
Recently, with advancements in AI, the integration of AI in level design has increasingly become a focal point of interest. Guzdial et al. [33] designed an AI-driven game level editor named "*Morai Maker*" for *Super Mario Bros.* to design style games and gave a deep insight into the relationship between AI and level designers. The integration of AI into game level editors may revolutionize how users interact with and create content in games, since AI can act as a co-creator, offering suggestions, optimizing design elements, and even generating entire levels based on minimal input from users, or analyze a player's gameplay style and automatically generate levels that challenge and engage them based on their unique play patterns [34, 87]. For both novice and experienced users, AI can make level customization much easier, even if there are issues with the fairness of creation and the potential uncontrollability of **AI-Generated Content (AIGC)** [15, 35].

5.2 Resource Editors

Unlike the original classification, we introduce the concept of "resource editor," because the resource files that can be changed or generated by users in video games also fall under the category of UGC. Generally speaking, we classify the resource editors into two directions, including object editors and plugin editors.

5.2.1 Object Editors. Object editors are the most common game editors. Our conference version [22] overlooked the presence of functions that allow customization of game objects in certain video games, which we now classify as object editors, even though they are integrated into the game itself. Role-playing games often feature fully functional and detailed character editors, but sometimes the default appearance of the character is not satisfactory to all players. Moreover, for the main character that the player primarily controls, if the developer can provide the corresponding editor, it can

[46]https://levelsharesquare.com/.
[47]https://levelsharesquare.com/games/supermarioconstruct.



(a) Built-in object editor (b) Unofficial texture editor

Fig. 13. Comparison of built-in and unofficial object editors.

solve this contradiction to a certain extent. *Cyberpunk 2077*[48] features a wide range of customizable character traits, from gender to body type to facial features, that can be modified through the in-game character editor (Figure 13(a)). These official object editors sometimes even indirectly affect the entire game play experience, such as *Sims 4* mentioned in Section 4.4. To this end, the functions of the in-game editor may also become one of the factors to evaluate the quality of the game.

In addition to official in-game editors, there are fully functional unofficial modification tools. For instance, *Tynker's texture pack editor*[49] (Figure 13(b)) is an unofficial object editor for *Minecraft* that allows users to edit pixels on the website to change different game items, character skins, and blocks. In contrast to the official editor, the resource files generated in the unofficial editor usually need to be manually imported into the game by the user. Moreover, compared with other types of editors, object editor essentially adds or replaces the original resource file (such as creating a new character model file) to achieve the purpose of modification, so for object editors, the difference between official and unofficial is not that obvious. Unofficial developers typically face challenges in accessing core game files and are often restricted by official developers from altering gameplay mechanics. The use of object editors is generally similar to model building, and in terms of functionality, general editors such as *Blender* can also achieve sophisticated modeling. Using external modeling software to create more detailed models for importing into a game is common practice. For instance, many user-made mods in *Sims 4* are created this way. In *Roblox*, these externally imported 3D models are known as "meshes," although players can also create models directly in *Roblox Studio*, the advantage of using external general editors lies in their greater professionalism and power.

5.2.2 Plugin Editors. Plugin editor is aimed to generate or manage plugins that can be applied in the game. The definition of the plugins has been described in Section 4.3, which corresponds to the user-generated instrument. Even if we classify plugins as a type of resource file, the plugin editor is much more complex than object editor, and usually involves accessing the core file and modifying some data of its settings. So far, the plugin editor only accounts for a small proportion of UGC editors, perhaps because creating plugins that can be applied to games requires users to have a certain programming background, and modifying game values also needs to be done through common Integrated Development Environment such as *Visual Studio*.[50] However, compilers that

[48] https://www.cyberpunk.net/hk/en/.
[49] https://www.tynker.com/minecraft/items/edit/.
[50] https://visualstudio.microsoft.com/.

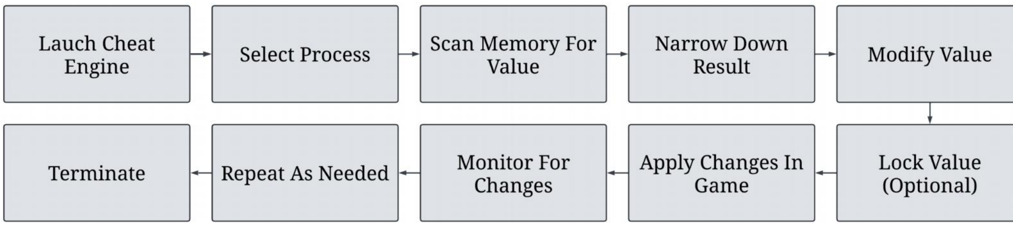


Fig. 14. A simplified flowchart of how *Cheat Engine* works when modifying a game's memory.

can be used to write plugin code files cannot be considered as plugin editors. We believe that the UGC editor should be distinguished from the tools used by professional developers.

Cheat Engine[51] is a modification tool for single-player games without internet connection. It allows users to modify some game parameters (such as health and damage) according to their needs. Equipped with a memory scanner, *Cheat Engine* enables users to find the correct variable and make numerical changes to debug them by quickly scanning the code files, where Figure 14 illustrates a simplified flowchart of modifying a game's memory. For example, given that the initial **Health Points (HP)** was 100 and the health was reduced by 1 after a hit. The *Cheat Engine* will scan for a variable with a value of 100, and then use the simulated "hit me" function in *Cheat Engine* to look again for a variable with a value of 99, which could narrow down the variables that actually point to HP so that other following changes of value can be made.

TES5Edit[52] is another special case of plugin editor used to operate and manage *The Elder Scrolls V: Skyrim*[53] plugins. In addition to generating plugins, its main function is to compare multiple plugins and find potential conflicts to achieve efficient management. There may be incompatibilities between different mods, but for players who have multiple mods installed in the same game, checking one by one is quite a trouble and can even reduce the player's interest in the game. Therefore, tools such as "module manager" can be used to solve this problem, and *TES5Edit* can also be used to remove official DLC that may cause the program to crash.

However, *Cheat Engine* is only available for offline single-player games and may not workable for online multiplayer games. Both the above examples are editors for single-player games, while in fact, the usage of plugins in multiplayer games, especially competitive multiplayer games, is an issue of fairness in the game, which is why developers in competitive games often openly prohibit cheating in the game. Online video games involve complex data transfers between servers and untrusted clients [54], making it harder for plugin developers to acquire code and make changes. Besides, multiplayer games typically feature dedicated anti-cheat systems or administrators tasked with detecting and preventing cheating, with significant penalties for those caught violating the rules. Despite the ongoing issues with cheating in popular FPS games like *Valorant*,[54] cheating plugins and the possible editors used to create them are generally not publicly available due to the strong opposition from game developers and the implementation of robust anti-cheat measures.

5.3 Lessons Learned

Based on the studies reviewed in Section 5, several broader insights regarding UGC editors can be identified: (1) UGC editors function not merely as technical tools but as mediators of creativity: their design decisions (such as modularity, interface complexity, and integration with game

[51] https://www.cheatengine.org/index.php.
[52] https://github.com/TES5Edit/TES5Edit/releases/.
[53] https://elderscrolls.bethesda.net/en/skyrim10.
[54] https://playvalorant.com/.

mechanics) directly influence what kinds of content users are likely to produce. (2) A recurring theme is the tradeoff between expressive power and accessibility. Editors with rich functionalities enable highly sophisticated creations but often exhibit steep learning curves, whereas lightweight or template-based editors promote mass participation but constrain creative diversity. (3) Most existing works emphasize editor functionality while overlooking user interaction processes, including learning behaviors, collaborative creation, and the evolution of user expertise over time. (4) Well-integrated in-game editors tend to foster stronger creator ecosystems than external standalone tools, suggesting that seamless embedding within gameplay significantly enhances user motivation and long-term engagement. (5) The literature reveals a lack of systematic evaluation methods for UGC editors: few studies assess the usability, creative productivity, or community impact of these tools. Overall, these findings highlight the need for future research to focus on user-centered editor design, longitudinal usage patterns, and the broader socio-technical dynamics that shape creation practices within UGC ecosystems.

6 Validation of the Classification Framework

To ensure that the proposed UGC classification is conceptually robust and practically applicable, we incorporated a structured expert review to confirm the clarity, completeness, and usability of the framework across diverse game contexts. We invited three domain experts with research experience or industrial experience in game design and interactive multimedia to provide feedback on the classification. The review materials included: (1) the proposed taxonomy with definitions; (2) its hierarchical structure; and (3) representative UGC and UGC editor cases. Experts were asked to evaluate the framework in terms of conceptual clarity, category boundaries, coverage, and applicability to real-world examples.

6.1 Key Insights From Expert Feedback

The informal conversations with domain colleagues yielded several helpful observations about the clarity and usefulness of the proposed classification:

— They agreed that separating UGC into the two broad categories of gameplay expansion and resource enhancement reflects how UGC is typically understood within both academic and industrial communities. This structure was described as intuitive and easy to communicate, especially when accompanied by concrete examples from well-known games.
— They noted that the boundaries between some subcategories could be subtle for readers who are not deeply familiar with UGC practices. Their comments encouraged us to be more explicit about the underlying rationale behind each category (e.g., distinguishing whether a piece of UGC primarily alters player decision-making or primarily modifies audiovisual presentation). This led us to refine some explanatory sentences and clarify the intent behind a few borderline cases.
— They emphasized that overlap across categories is normal in game modification culture. Several colleagues pointed out real examples from their own experiences (e.g., mods that simultaneously change mechanics and assets), reinforcing that a flexible approach based on a primary-effect perspective is more practical than enforcing strict exclusivity. These discussions helped shape the lightweight overlap-guidance principles adopted in this section.

Overall, the informal feedback confirmed that the framework is understandable and aligns with how UGC is commonly recognized in practice, while also highlighting small areas where clearer wording could enhance accessibility for a broader readership.

6.2 Addressing Category Overlaps

As UGC is inherently multifaceted, certain creations may naturally span multiple categories (e.g., a mod that introduces both new gameplay rules and new character assets). Following established practices in classification research, our aim is not to eliminate overlap, but to ensure consistent and interpretable categorization. To this end, we propose three potential guidelines to better solve the issue of overlaps:

— *Primary Functional Intent*. A piece of UGC is classified according to the change that most substantially alters the player's experience. For example, if a mod introduces both new assets and new mechanics, the category is determined by which aspect has the greater functional impact during gameplay.
— *Player-Facing Interpretation*. When the creator's intent is ambiguous, the classification follows how players predominantly encounter and describe the content in practice. In other words, we prioritize the perspective of the end-user community since UGC ultimately takes effect through player perception and interaction.
— *Creation-Tool and Workflow Clues*. If uncertainty still remains, the design and affordances of the editor or tool used to create the UGC serve as an additional indicator. Tools oriented toward scripting, rules, or logic tend to produce gameplay-expansion content, whereas tools focused on assets, modeling, or interface components commonly yield resource-enhancement UGC. These workflow cues offer practical guidance for resolving borderline cases.

These lightweight principles allow overlapping or hybrid cases to be handled at a certain degree without imposing rigid or fixed boundaries, maintaining both reliability and flexibility in real-world application.

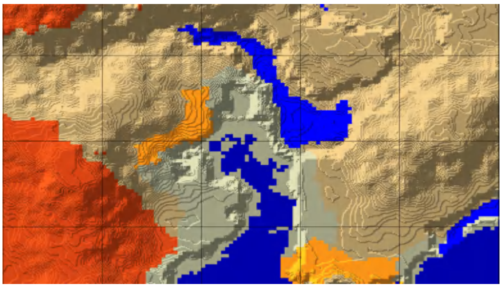
7 Discussion

In the previous sections, we completed the classifications of UGC and UGC editors and analyzed relevant examples. However, there still exists limitations in our study. The following subsections will address the issues raised above and outline prospects for future research.

7.1 Application of AI in UGC

As an advanced form of PCG, AIGC has recently been applied in various games, particularly in character design and resource generation [82, 83, 96]. Currently, AIGC mainly uses deep learning methods, which can be more powerful and flexible than traditional level or map generation methods [83]. For instance, Mi et al. [63] used the Black bear monster from "Journey to the West" to introduce an application path of AIGC in actual game character design.

We can utilize *Minecraft*'s world generation as an illustrative example: The traditional method involves providing a seed (typically a string of limited length) through a specific algorithm, generates a pseudo-random value (as computers inherently struggle with generating true randomness) [5]. Therefore, this value of the seed determines the terrain and features of the world. In fact, the process of manually entering the value of the seed in the game to enjoy the random generation of the world can also be regarded as a kind of UGC creation. However, the seed is not directly related to the actual generated world. On the one hand, since the block updates and species generation are dictated by the developer's algorithm, the same seed will consistently produce the same world. On the other hand, a limitation of this approach is that slight variations in the algorithm across different game versions can lead to unpredictable differences in the terrain and landscapes generated within the game. For example, a seed named "rainforest" does not mean that it can generate a world with rainforest landforms.



(a) Seed-Generated World

(b) AI-Assisted Dune City Created By Kaizen87

Fig. 15. Comparison of World Generation in *Minecraft*.

To meet the requirements, users can choose to customize each block with coordinates in some unofficial map editor such as *MCEdit*,[55] but setting blocks one by one can be inefficient, because in the classic version of *Minecraft* there are approximately 1.38 trillion blocks per world [43]. Instead of relying on PCG for world creation, AI can analyze real-world photos from the internet and transform them into pixel-style 3D models. This allows for the generation of worlds that differ significantly from the basic natural terrain in *Minecraft*, enabling the creation of environments like modern cities. Recent years some AI-powered generators such as *BuilderGPT*[56] developed by Zhou-Shilin have been adopted by users to assist to generate more creative maps, making the creation process automatic or semi-automatic. Moreover, Awiszus et al. [3] proposed a generative model called "World-GAN" for *Minecraft* worlds, which is based on NLP and is applied to generating larger and more complex worlds while meeting the needs of specified terrain. Figure 15 shows a comparison between traditional generation and AI-assisted generation.

While AI simplifies the most time-consuming part and reduces the cost of designing and creating new worlds for users, it is not entirely satisfactory at generating content that meets players' expectations, especially text-to-3D, which is almost unavailable [83]. However, the relevant technology is still evolving, and we believe that the future trend of UGC must be inseparable from the dialectical view of AIGC. As we mentioned in Section 5.1.2, the use of AI as a supplementary tool can not only lower the creation barriers but also enhance users' creative enthusiasm, since its role is particularly evident in processing large amounts of data or automating repetitive tasks. While the rise of AIGC may involve challenges such as copyright issues and cultural conflicts [31], we suggest that the potential benefits of AIGC outweigh these concerns if properly handled. It is crucial to deeply consider how to responsibly manage the application of AI in UGC creation.

Overall, emerging AI-assisted tools are transforming user creation from manual construction to intelligent collaboration. AI no longer merely automates tasks, while it amplifies human creativity and reshapes the production loop between users and games. This evolution marks a many shifts from passive participation to active co-creation within the UGC ecosystem:

- *Integration of UGC and AI: From Automation to Co-Creation.* The integration of AIGC into UGC creation extends beyond procedural generation toward *co-creation*. AI systems embedded in editors can interpret user prompts and creative intent, providing adaptive feedback and content suggestions. This shifts the workflow from manual construction to interactive collaboration, in which players guide AI tools iteratively to refine design outcomes. Such human–AI synergy expands creative possibilities while preserving user agency.

[55] https://www.mcedit.net/.
[56] https://github.com/CubeGPT/BuilderGPT.

- *Lowering the Barrier to Creation.* AI substantially reduces the technical threshold for participation by automating complex steps such as modeling, texturing, or scripting. Through natural-language instructions or example-based generation, non-expert players can focus on story, style, and gameplay experience instead of coding. This democratization encourages diverse participation and promotes a sustainable long-tail ecosystem of creators contributing at different scales.
- *AI-Assisted Evaluation and Feedback.* Beyond generation, AI can assist in evaluating and refining user-created works. By analyzing gameplay data and interaction patterns, machine learning models can identify balance issues, predict engagement, or suggest content improvements. Such adaptive feedback mechanisms help maintain quality within large UGC ecosystems and close the loop between creation and optimization.
- *Community-Level Impact and Recommendation.* AI-driven recommendation systems are increasingly used to organize and promote UGC within player communities. Intelligent curation can surface creative or niche works that might otherwise be overlooked, fostering diversity in exposure. However, understanding how algorithmic ranking influences user motivation, visibility, and perceived fairness remains an open research challenge in participatory game culture.

7.2 Ethical Consideration

7.2.1 Cheating Phenomenon. Multiplayer online games often raise ethical concerns regarding cheating, where cheaters employ software robot or manipulate execution code to gain an unfair advantage over other players [54, 97]. Despite the implementation of strict anti-cheating measures, including detection systems and severe penalties to deter such behavior, the allure of potential fame and fortune continues to drive a significant number of users to cheat. Simultaneously, for game developers, the implementation of anti-cheating mechanisms and the enforcement of penalties for cheaters significantly increased the development and maintenance costs of the game [6], thereby exacerbating the conflict between players and the industries. In academia, Yan and Randell [95] proposed a classification of cheating in online games in 2005, considering that fairness is a crucial factor influencing the prevalence of cheating. Consalvo [14] defined cheating in both multiplayer and single-player games, which noted that many players believed using external tools to play a game after completing it was not cheating, though this view was generally confined to single-player experiences (which we define as “user-generated instruments” in Section 4.3). Broadly speaking, cheating involves the player’s actions that break the game’s rules. However, when discussing cheating, the primary concern is fairness in multiplayer games. In these contexts, cheaters gain a significant advantage over other players, disrupting the game’s balance [2]. On an ethical level, Szell and Thurner [89] argued that the competitive and interactive nature of games mirrored real-world dynamics, suggesting that widespread cheating in games reflects a broader lack of moral awareness among players in society.

When it comes to UGC, if cheating is simply defined as breaking certain rules within the game, much of the UGC discussed earlier could be classified as cheating. Unlike third-party guides or tutorials that for instance help players obtain hidden achievements, cheating involves circumventing the game’s rules to gain an unfair advantage that other players cannot access. We notice that tools often considered cheating, such as auto-targeting modifiers, are also user-generated and a form of creativity [36]. Blackburn’s research on Steam’s user community [6] suggests that though cheaters are largely indistinguishable from fair players, that users who are labelled as cheaters suffer a “social penalty” of losing friends. Ethically, the primary harm of cheating lies in its disruption of game balance and the subsequent negative impact on the gaming experience, as well as the game’s lifecycle. When discussing how to prevent the generation of cheat-related content, developing moral negative feedback mechanisms from a social and group psychological perspective may be a viable solution [26].

7.2.2 Copyright Infringement. In the process of user generation, copyright infringements caused by weak copyright awareness are not rare. We categorize these conflicts into two main types. (1) The conflict between users and third-party copyright owners: The most obvious example is, when users (also noted as modders in some articles) introduce resource files from third-party products without permission, players who do not own those products can access the resources by downloading the relevant modification of the game. Additionally, when AI is employed as a tool in user generation, it often involves the use of unauthorized works during the model training process [21, 24]. On the other hand, there are someone believes that creators who make mods for non-profit purposes should not have to pay a legal price for the freedom to create [74]. (2) The conflict between users and game developers: Some game industries, such as *Atari*, have argued that games are their own property, and that any content created based on them is infringing [37]. While there is not yet a game that has completely rejected UGC, it is common for those game developers to limit the scope of what UGC can do, dividing the game into a part that users can change and a core that cannot be changed. Not only for UGC that changes the game itself, but also for media-based UGC such as game livestreams on *YouTube* or related game-related videos are at risk of being sued by developers for infringement [47].

Legally speaking, video games are more complex than other copyrighted products, while many local laws treat video games as computer programs [1]. However, as video games have evolved, people have come to realize that the resource files (images and sounds) and gameplay in video games also need copyright protection, which is still controversial. For example, most FPS games on the market are basically the same in terms of gameplay. Copyright infringements are essentially conflicts of interest. Giving priority to the interests of the developer or copyright owner, while limiting the authorities of the user to create, stifles user's creativity and expression, and deprives the community of a way to benefit from the platform [74]. Regarding the solution of copyright problem in video games, we agree with Jungar's [47] view that the purpose of copyright is not to "maximize revenue," but to "incentivize innovation." Taking a negative attitude toward creators and completely denying the rationality of the existence of UGC are not conducive to the copyright owners to obtain more profits.

7.3 Game, User, and Community

To envision the future of UGC in the video game industries, it is essential to understand the intricate relationship between video games, users, and communities [73]. Figure 16 presents a schematic diagram that visualizes the dynamic interaction between the three entities and highlights how UGC acts as a bridge, enhancing engagement and boosting creativity within video game industry.

Sánchez et al. [81] analyzes the user experience in video games and puts forward the concept of playability, which proves that users consume video games. As an excellent test bed for HCI, users play a dominant role in the game design, which is called user-centered design [69]. Since user is often referred to as the player or more generally the consumer, user experience is often the fundamental criterion for evaluating the quality of the game. Adam [2] argues in his book that games have certain reward and punishment mechanisms that incentivizes sustained interest in the game. For example, in *PUBG: Battlegrounds*,[57] players who get killed cannot continue to participate in the competition, and the last player or team that survives will win the game. This is a typical reward and punishment mechanism, where the punishment does not mean that the player has broken the rules, but rather gives the player frustration through social comparison from a

[57] https://www.pubg.com/.

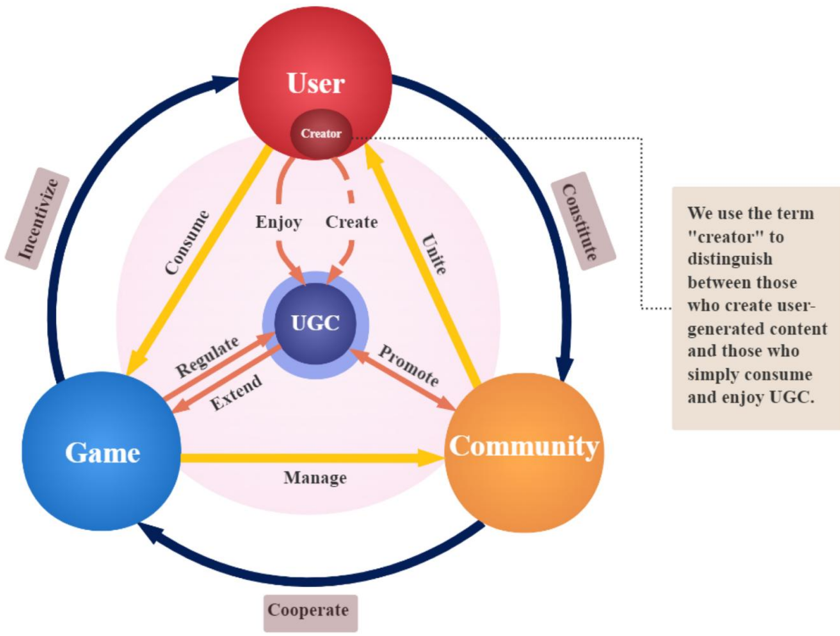


Fig. 16. The relationship graph between video game, user, and community, including the connections between the three and UGC.

psychological point of view, similarly, the winners will reward positive emotions such as happiness [30]. Successful game designers use rewards and punishments to incentivize users and increase user engagement, thereby extending the lifecycle of the game.

Communities unite users around the world, while the social behaviors among users constitute communities. As a third party group that does not directly consume video games, community has often been regarded as a platform for users to share and communicate in past studies [80]. Moreover, Hsu [41] recognizes the role of online communities in increasing customer loyalty and differentiates business communities from entertainment communities. Some communities like *Planet Minecraft*[58] are created solely for specific games, while others, such as *Nexus mods*,[59] exist as a popular platform for sharing mods. With the development of network, the function of the community has gradually expanded to provide a platform for trading in-game items and other goods. The community usually extracts part of the profits from users' transaction to maintain the daily operation of the website. Besides, advertising, fees paid by a few privileged users, and so on, also bring revenue to the community. In turn, the community inspires a sense of belonging among users, and the events and competitions held within the community increase user participation and provide emotional support to foster social ties [90].

Game community is closely related to video games themselves [10], since community is one of the quickest ways to give developers timely feedback on game evaluation and user experience. While some game developers create and manage communities as spin-offs of video games, unofficial communities are not affiliated with any game. Even though most of those third-party communities claim to be non-profit, there are significant costs to running and maintaining the site, while, in fact, the community benefits from traffic and user attention. As we mentioned in Section 7.2, there

[58] https://www.planetminecraft.com/.
[59] https://www.nexusmods.com/.

is a potential conflict of interest between the community and the video game developer. However, when it comes to the relationship between video games and the community, we believe that official intervention to manage the community on a cooperative basis, where necessary, might be the best solution. For example, game developers have the right to intervene and police the community, where there is an interesting phenomenon within the communities that they encourage the exploitation of game bugs to gain an unfair competitive advantage.

The development of video games has benefited a lot from UGC [61], and developers should have a positive attitude toward UGC considering about the prospect of UGC and video games, by regulating UGC appropriately, or integrating UGC into original games (e.g., *Roblox*), and relying on the openness of UGC to drive higher monetization. In terms of content, UGC, as a derivative of video games, is intended to expand the vacancy not covered in the game. Daugherty's research on consumer motivation [17] suggests that UGC helps users alleviate self-doubt, while Xu et al. [94] argues that UGC serves as a vital medium for players to express their emotions while gaming. In this study, we define "creators" as users who possess creative abilities and actively engage in producing UGC. These creators have the ability to enjoy and consume UGC that all users have. Within the community, UGC plays a crucial role in fostering a sense of belonging and driving its long-term development of games [49, 88].

7.4 Overlooked Role of Fandom of Video Game Culture

The UGC discussed in this article mainly focuses on content that directly impacts the original game, with our study examining how UGC alters the game itself. However, it is also important to acknowledge the existence of UGC that extends beyond the game itself, resulting in derivative works such as videos, fan fiction, and other creative outputs inspired by the game's characters and storylines. In recent years, these secondary creations based on existing works have been referred to as "fandom." As a part of popular culture, fandom by definition fits into Wirman's [93] five dimensions of player productivity as "productivity beyond play: expressive productivity," while the UGC we studied in this article belongs the "productivity for play: instrumental productivity" [28]. Moreover, Jones [45] believes that video games provide a great platform for studying fan culture. Similarly, some researchers have pointed out that video game is a representation of participatory culture [19, 55, 74, 77], where fans' creative contributions can create many different forms in the video games.

Since this part of UGC falls outside the scope of our classification, we acknowledge that the classification system we propose may not be fully comprehensive. Regarding the fandom of video culture, there are some studies that have provided detailed discussions, so we list a part of representative works for reference. For example, Dym et al. [25] pointed out that fan-created communities can broaden the scope of video game culture, introducing more gender diversity in storytelling and challenging hegemonic barriers. Pearce [71] argued that fandom has expanded the culture of video games, as fans not only consume content but also produce, modify, and distribute their creations. Booth [7] considered that fan-created content also influenced the perception and experience of games themselves. In fact, as an indispensable part of UGC, the influence of fan creation on video game culture and communities continues to grow [32]. In future research, exploring the relationship between fan creations and video games, as well as their impact, will be highly valuable for both industry and academia.

7.5 Relationship Between Game Genre and User Creative Tendencies

Our study suggests a potential link between the type of game and the tendency of users to create UGC. For instance, in sandbox games like *Minecraft*, UGC frequently takes the form of resource files, whereas in adventure games such as the *Mario series*, users tend to create new levels or campaigns.

The tendency toward UGC seems to be related to the convenience that the games provide for the creation of corresponding genres. Additionally, the attitudes of game developers toward UGC appear to influence users' willingness to engage in content creation. Daugherty et al. [17] pointed out that incentives related to content creation may significantly impact user motivation. However, existing research on UGC creation has predominantly focused on user-driven factors, leaving a gap in academic inquiry regarding the relationship between game types and user motivation for UGC creation. Therefore, we also consider that game type should be regarded as an external drive influencing users' propensity to create UGC, with different game genres potentially shaping different characteristic of the UGC produced.

8 Conclusion

In recent years, the development of video games and UGC has become increasingly intertwined, with UGC emerging as a crucial element in the future of the game industry. Based on previous studies, this article refines the existing classification of in-game UGC and corresponding editors, addressing inconsistencies and ambiguities in earlier categorizations. In the discussion, we highlight the importance of UGC outside games that belongs to fandom content in game culture, which is not covered in the classification. We also speculate on the potential relation between game genres and the tendency of users to create UGC, providing insights into the impact of AI on user generation. In addition, ethical concerns surrounding cheating and copyright disputes in gaming are explored, with proposed solutions based on relevant literature. Finally, we examine the relationship between games, users, communities, and UGC. We believe that the growth of UGC is positively correlated with the overall prospects of the video game industry, and the prosperity and innovation of UGC are poised to drive profitability within the game industry and academic research.

References

[1] Mr Stan Abrams. 2013. The legal status of video games: Comparative analysis in national approaches. *WIPO Magazine*, 7–96.

[2] Ernest Adams. 2014. *Fundamentals of Game Design*. Pearson Education.

[3] Maren Awiszus, Frederik Schubert, and Bodo Rosenhahn. 2021. World-GAN: A generative model for minecraft worlds. In *Proceedings of the 2021 IEEE Conference on Games (CoG)*. IEEE, 1–8.

[4] Albert Barreda and Anil Bilgihan. 2013. An analysis of user-generated content for hotel experiences. *Journal of Hospitality and Tourism Technology* 4 (2013), 263–280.

[5] Jens Bergensten. 2011. A short demystification of the map seed. Retrieved from https://web.archive.org/web/20120307165955; https://www.mojang.com/2011/02/a-short-demystification-of-the-map-seed/

[6] Jeremy Blackburn, Nicolas Kourtellis, John Skvoretz, Matei Ripeanu, and Adriana Iamnitchi. 2014. Cheating in online games: A social network perspective. *ACM Transactions on Internet Technology* 13, 3 (2014), 1–25.

[7] Paul Booth. 2015. *Game Play: Paratextuality in Contemporary Board Games*. Bloomsbury Publishing.

[8] Adam Broadbent and Andrea Corradini. 2024. Using genetic algorithms for the generation of increasingly challenging terrain for players to navigate. In *Proceedings of the 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 1–8.

[9] Christopher S. Brunt, Amanda S. King, and John T. King. 2020. The influence of user-generated content on video game demand. *Journal of Cultural Economics* 44, 1 (2020), 35–56.

[10] Thierry Burger-Helmchen and Patrick Cohendet. 2011. User communities and social software in the video game industry. *Long Range Planning* 44, 5–6 (2011), 317–343.

[11] Loïc Caroux. 2016. Low-level visual features of video game interfaces and player expertise. In *Actes de la 28ième Conference Francophone Sur L'Interaction Homme-Machine*, 126–133.

[12] George Christodoulides, Colin Jevons, and Jennifer Bonhomme. 2012. Memo to marketers: Quantitative evidence for change—How user-generated content really affects brands. *Journal of Advertising Research* 52, 1 (2012), 53.

[13] Francesca Comunello and Simone Mulargia. 2015. User-generated video gaming: Little big planet and participatory cultures in Italy. *Games and Culture* 10, 1 (2015), 57–80.

[14] Mia Consalvo. 2009. *Cheating: Gaining Advantage in Videogames*. MIT Press.

[15] Michael Cook and Simon Colton. 2014. Ludus ex machina: Building a 3D game designer that competes alongside humans. In *Proceedings of the ICCC*, 54–62.

[16] Maria Teresa Cuomo, Debora Tortora, Alex Giordano, Giuseppe Festa, Gerardino Metallo, and Erika Martinelli. 2020. User-generated content in the era of digital well-being: A netnographic analysis in a healthcare marketing context. *Psychology & Marketing* 37, 4 (2020), 578–587.

[17] Terry Daugherty, Matthew S. Eastin, and Laura Bright. 2008. Exploring consumer motivations for creating user-generated content. *Journal of Interactive Advertising* 8, 2 (2008), 16–25.

[18] Myriam Davidovici-Nora. 2009. The dynamics of co-creation in the video game industry: The case of world of warcraft. *Communications & Strategies* 73 (2009), 43.

[19] Christy Dena. 2008. Emerging participatory culture practices: Player-created tiers in alternate reality games. *Convergence: The International Journal of Research into New Media Technologies* 14, 1 (2008), 41–57.

[20] Rosa Lilia Segundo Díaz, Gustavo Rovelo Ruiz, Miriam Bouzouita, and Karin Coninx. 2021. Building blocks for creating enjoyable games—A systematic literature review. *International Journal of Human-Computer Studies* 159 (2021), 102758.

[21] Haihan Duan, Abdulmotaleb El Saddik, and Wei Cai. 2024. Incentive mechanism design toward a win–win situation for generative art trainers and artists. *IEEE Transactions on Computational Social Systems* 11 (2024), 7528–7540.

[22] Haihan Duan, Yiwei Huang, Yifan Zhao, Zhen Huang, and Wei Cai. 2022. User-generated content and editors in video games: Survey and vision. In *Proceedings of the 2022 IEEE Conference on Games (CoG)*. IEEE, 536–543.

[23] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai. 2021. Metaverse for social good: A university campus prototype. In *Proceedings of the 29th ACM International Conference on Multimedia*, 153–161.

[24] Haihan Duan, Tengfei Ma, Yuyang Qin, Runhao Zeng, Wei Cai, Victor C. M. Leung, and Xiping Hu. 2025. DeRelayL: Sustainable decentralized relay learning. *IEEE Transactions on Mobile Computing* 24 (2025), 8913–8929.

[25] Brianna Dym, Jed Brubaker, and Casey Fiesler. 2018. "Theyre all trans sharon": Authoring gender in video game fan fiction. *Game Studies* 18, 3 (2018), 1–28.

[26] Naomi Ellemers and Kees van den Bos. 2012. Morality in groups: On the social-regulatory functions of right and wrong. *Social and Personality Psychology Compass* 6, 12 (2012), 878–889.

[27] Antonia Estrella-Ramón and Fiona Ellis-Chadwick. 2017. Do different kinds of user-generated content in online brand communities really work? *Online Information Review* 41 (2017), 954–968.

[28] Teresa Maria Leal Ferreira. 2022. *Fandom, Culture, and Videogames: Analysing What Makes a Successful Videogame Franchise Through Final Fantasy*. Ph.D. Dissertation. Instituto Politecnico do Porto (Portugal).

[29] William Forsyth. 2016. Globalized random procedural content for dungeon generation. *Journal of Computing Sciences in Colleges* 32, 2 (2016), 192–201.

[30] Stephen M. Garcia, Avishalom Tor, and Tyrone M. Schiff. 2013. The psychology of competition: A social comparison perspective. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science* 8, 6 (2013.) 634–650.

[31] Daniel J. Gervais. 2019. The machine as author. *Iowa Law. Review* 105 (2019), 2053.

[32] Jonathan Gray, Cornel Sandvoss, and C. Lee Harrington. 2017. *Fandom: Identities and Communities in a Mediated World*. NYU Press.

[33] Matthew Guzdial, Nicholas Liao, Jonathan Chen, Shao-Yu Chen, Shukan Shah, Vishwa Shah, Joshua Reno, Gillian Smith, and Mark O. Riedl. 2019. Friend, collaborator, student, manager: How design of an AI-driven game level editor affects creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13.

[34] Matthew Guzdial and Mark Riedl. 2016. Game level generation from gameplay videos. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 12, 44–50.

[35] Matthew Guzdial and Mark Riedl. 2018. Automated game design via conceptual expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 14, 31–37.

[36] Karla R. Hamlen and Fran C. Blumberg. 2015. Problem solving through "cheating" in video games. In *Video Games and Creativity*. Garo P. Green and James C. Kaufman (Eds.), Elsevier, 83–97.

[37] Thomas Hemnes. 1982. Adaptation of copyright law to video games. *University of Pennsylvania Law Review* 131 (1982), 171.

[38] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 9, 1 (2013), 1–22.

[39] Andrew Hicks, Veronica Cateté, and Tiffany Barnes. 2014. Part of the game: Changing level creation to identify and filter low quality user-generated levels. In *Proceedings of the FDG*.

[40] Drew Hicks, Zhongxiu Liu, Michael Eagle, and Tiffany Barnes. 2016. Measuring gameplay affordances of user-generated content in an educational game. In *Proceedings of the International Conference on Educational Data Mining*. International Educational Data Mining Society.

[41] Chin-Lung Hsu and Hsi-Peng Lu. 2007. Consumer behavior in online game communities: A motivational factor perspective. *Computers in Human Behavior* 23, 3 (2007), 1642–1659.

[42] Daniel Hufnal, Ethan Osborne, Theodore Johnson, and Caglar Yildirim. 2019. The impact of controller type on video game user experience in virtual reality. In *Proceedings of the 2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–9.

[43] Robert Earl Wells Iii. 2022. How Big Is a Minecraft World? Retrieved from https://www.lifewire.com/how-big-is-a-minecraft-world-5212822

[44] Diana Ipacs. 2024. Level Design: What Is It and What Makes It Successful? Retrieved from https://bluebirdinternational.com/level-design/

[45] Robert Jones. 2006. From shooting monsters to shooting movies: Machinima and the transformative play of video game fan culture. In *Fan Fiction and Fan Communities in the Age of the Internet: New Essays*. Karen Hellekson and Kristina Buss (Eds.), 261–80.

[46] Young Joo Kang, Ui Jun Lee, and Saerom Lee. 2024. Who makes popular content? Information cues from content creators for users' game choice: Focusing on user-created content platform "Roblox". *Entertainment Computing* 50 (2024), 100697. DOI: https://doi.org/10.1016/j.entcom.2024.100697

[47] Eirik Evert Elias Jungar. 2016. Streaming video games: Copyright infringement or protected speech? *Press Start* 3, 2 (2016), 22–47.

[48] Anssi Kanervisto, Tomi Kinnunen, and Ville Hautamäki. 2022. GAN-Aimbots: Using machine learning for cheating in first person shooters. *IEEE Transactions on Games* 15, 4 (2022), 566–579.

[49] Vlasios Kasapakis and Damianos Gavalas. 2017. User-generated content in pervasive games. *Computers in Entertainment* 16, 1 (2017), 1–23.

[50] Ahmed Khalifa, Diego Perez-Liebana, Simon M. Lucas, and Julian Togelius. 2016. General video game level generation. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 253–259.

[51] Yubo Kou and Xinning Gui. 2023. Harmful design in the metaverse and how to mitigate it: A case study of user-generated virtual worlds on Roblox. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, 175–188.

[52] Mark Kretzschmar and Mel Stanfill. 2019. Mods as lightning rods: A typology of video game mods, intellectual property, and social benefit/harm. *Social & Legal Studies* 28, 4 (2019), 517–536.

[53] Greg Lastowka. 2007. User-generated content and virtual worlds. *Vanderbilt Journal of Entertainment & Technology Law* 10 (2007), 893.

[54] Peter Laurens, Richard F. Paige, Phillip J. Brooke, and Howard Chivers. 2007. A novel approach to the detection of cheating in multiplayer online games. In *Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS '07)*. IEEE, 97–106.

[55] Qing Li. 2010. Digital game building: Learning in a participatory culture. *Educational Research* 52, 4 (2010), 427–443.

[56] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. 2013. Sentient sketchbook: Computer-assisted game level authoring. In *Proceedings of the 8th International Conference on the Foundations of Digital Games.*

[57] Zhonghao Lin, Haihan Duan, Zikai Alex Wen, and Wei Cai. 2022. What features influence impact feel? A study of impact feedback in action games. In *Proceedings of the 2022 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–6.

[58] Simon Liu, Li Chaoran, Li Yue, Ma Heng, Hou Xiao, Shen Yiming, Wang Licong, Chen Ze, Guo Xianghao, Lu Hengtong, et al. 2019. Automatic generation of tower defense levels using PCG. In *Proceedings of the 14th International Conference on the Foundations of Digital Games (FDG '19)*. ACM, New York, NY. DOI: https://doi.org/10.1145/3337722.3337723

[59] Amari Low, Jane Turner, and Marcus Foth. 2023. Worlds apart, together: Discovering players' placemaking priorities in cooperative terraforming games. In *Proceedings of the 35th Australian Computer-Human Interaction Conference*, 148–158.

[60] Michael Luca. 2015. User-generated content and social media. In *Handbook of Media Economics*. Simon P. Anderson, Joel Waldfogel, and David Strömberg (Eds.), Vol. 1. Elsevier, 563–592.

[61] Joakim Lundmark and Eric Sandström Lindberg. 2015. The effects of intrinsic motivation, extrinsic motivation and toolkits onuser participation in User-generated content for video games:: A quantitative study of product development in online communities. Retrieved from https://www.diva-portal.org/smash/get/diva2:827713/FULLTEXT01.pdf

[62] Marcello A. Gómez Maureira, Lisa E. Rombout, Livia Teernstra, Imara C. T. M. Speek, and Joost Broekens. 2015. The influence of subliminal visual primes on player affect in a horror computer game. In *Proceedings of the 2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 705–711.

[63] Xin Mi, Yang Cao, and Qiao Li. 2023. The application path of AIGC in assisting game character design—A case study of the black bear monster from journey to the west. In *Proceedings of the 2023 4th International Conference on Intelligent Design (ICID)*. IEEE, 275–282.

[64] Tian Min, Hanyi Wang, Yaoze Guo, and Wei Cai. 2019. Blockchain games: A survey. In *Proceedings of the 2019 IEEE Conference on Games (CoG)*. IEEE, 1–8.

[65] David Moher, Alessandro Liberati, Jennifer Tetzlaff, and Douglas G. Altman. 2009. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *BMJ* 339 (2009), b2535.

[66] Kyle Andrew Moody. 2014. *Modders: Changing the Game through User-Generated Content and Online Communities*. The University of Iowa.

[67] Teresa K. Naab and Annika Sehl. 2017. Studies of user-generated content: A systematic review. *Journalism* 18, 10 (2017), 1256–1273.

[68] Gary Ng, Joon Gi Shin, Alexander Plopski, Christian Sandor, and Daniel Saakes. 2018. Situated game level editing in augmented reality. In *Proceedings of the 12th International Conference on Tangible, Embedded, and Embodied Interaction*, 409–418.

[69] Yiing Y'ng Ng and Chee Weng Khong. 2014. A review of affective user-centered design for video games. In *Proceedings of the 2014 3rd International Conference on User Science and Engineering (i-user)*. IEEE, 79–84.

[70] Anh Nhu, Hieu Phan, Chang Liu, and Xianglong Feng. 2023. A comprehensive defense approach targeting the computer vision based cheating tools in FPS video games. In *Proceedings of the 2023 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 168–177.

[71] Celia Pearce. 2011. *Communities of Play: Emergent Cultures in Multiplayer Games and Virtual Worlds*. MIT Press.

[72] Lev Poretski and Ofer Arazy. 2017. Placing value on community co-creations: A study of a video game 'Modding' community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, 480–491. DOI: https://doi.org/10.1145/2998181.2998301

[73] Hector Postigo. 2007. Of mods and modders: Chasing down the value of fan-based digital game modifications. *Games and Culture* 2, 4 (2007), 300–313.

[74] Hector Postigo. 2008. Video game appropriation through modifications: Attitudes concerning intellectual property among modders and fans. *Convergence: The International Journal of Research into New Media Technologies* 14, 1 (2008), 59–74.

[75] Xin Qian, Rafet Sifa, Xuefei Liu, Shreyashi Ganguly, Borchuluun Yadamsuren, Diego Klabjan, Anders Drachen, and Simon Demediuk. 2022. Anomaly detection in player performances in multiplayer online battle arena games. In *Proceedings of the 2022 Australasian Computer Science Week (ACSW '22)*. ACM, New York, NY, 23–30. DOI: https://doi.org/10.1145/3511616.3513095

[76] J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning* 1 (1986), 81–106.

[77] Joost Raessens and Jeffrey Goldstein. 2005. Computer games as participatory media culture. In *Handbook of Computer Game Studies*. Joost Raessens, Jeffrey Goldstein (Ed.), 373–388.

[78] Nuno Ramos, Pedro Santos, and João Dias. 2023. Dual critic conditional Wasserstein GAN for height-map generation. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 1–4.

[79] Caroline Rendenbach and Daniel Dyrda. 2024. Proposing formal game spaces as foundation for gameplay-based anti-cheating measures. In *Proceedings of the 2024 IEEE Conference on Games (CoG)*. IEEE, 1–4.

[80] Howard Rheingold. 2000. *The Virtual Community, Revised Edition: Homesteading on the Electronic Frontier*. MIT Press.

[81] José Luis González Sánchez, Francisco Luis Gutiérrez Vela, Francisco Montero Simarro, and Natalia Padilla-Zea. 2012. Playability: Analysing user experience in video games. *Behaviour & Information Technology* 31, 10 (2012), 1033–1054.

[82] Jonathan Schaeffer and H. Jaap Van den Herik. 2002. Games, computers, and artificial intelligence. *Artificial Intelligence* 134, 1–2 (2002), 1–7.

[83] Zhuoheng Shen. 2023. Effects of AI-Generated Content (AIGC) in the Game Development: From traditional PCG to AIGC. University of Skövde. Retrieved from https://www.diva-portal.org/smash/get/diva2:1800740/FULLTEXT01.pdf

[84] Remi Smith, Tamara Fodora, and Itai Guzik. 2023. Dota 2 vs. World of Warcraft: What is the Difference? Retrieved from https://www.buff.game/a/blog/dota-2-vs-world-of-warcraft-what-is-the-difference

[85] Olli Sotamaa and Hanna Wirman. 2015. Online games modifications and user generated content. In *The International Encyclopedia of Digital Communication and Society*. Robin Mansell, Peng Hwa Ang, Charles Steinfield, Shenja van der Graaf, Pieter Ballon, Aphra Kerr, James D. Ivory, Sandra Braman, Dorothea Kleine, and David J. Grimshaw (Eds.), 1–10.

[86] Adam Summerville and Michael Mateas. 2016. Super Mario as a string: Platformer level generation via LSTMs. arXiv:1603.00930. Retrieved from https://arxiv.org/abs/1603.00930

[87] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. 2018. Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games* 10, 3 (2018), 257–270.

[88] Yacheng Sun, Xiaojing Dong, and Shelby McIntyre. 2017. Motivation of user-generated content: Social connectedness moderates the effects of monetary rewards. *Marketing Science* 36, 3 (2017), 329–337.

[89] Michael Szell and Stefan Thurner. 2010. Measuring social dynamics in a massive multiplayer online game. *Social Networks* 32, 4 (2010), 313–329.

[90] Tina L. Taylor. 2009. *Play between Worlds: Exploring Online Game Culture*. MIT Press.

[91] Poom Thiparpakul, Tanat Jiravansirikul, and Tuang Dheandhanoo. 2020. The satisfaction of sword's sound effect in game design in case of metal gear rising and Sekiro: Shadow dies twice. In *Proceedings of the 2020 8th International Conference on Information and Education Technology*, 252–256.

[92] Dandison C. Ukpabi and Heikki Karjaluoto. 2018. What drives travelers' adoption of user-generated content? A literature review. *Tourism Management Perspectives* 28 (2018), 251–273.

[93] Hanna Wirman. 2009. On productivity and game fandom. *Transformative Works and Cultures* 3 (2009), 8.

[94] Yong Xu, Dongqin Li, Fengshan Si, and Shuqin Huang. 2017. Study on sentimental analysis for intermedia-UGC. In *Proceedings of the 2017 2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE '17)*. Atlantis Press, 607–611.

[95] Jeff Yan and Brian Randell. 2005. A systematic classification of cheating in online games. In *Proceedings of the 4th ACM SIGCOMM Workshop on Network and System Support for Games*, 1–9.

[96] Georgios N. Yannakakis and Julian Togelius. 2018. *Artificial Intelligence and Games*, Vol. 2. Springer.

[97] Siu Fung Yeung, John Lui, Jiangchuan Liu, and Jeff Yan. 2006. Detecting cheaters for multiplayer games: Theory, design and implementation. In *Proceedings of the 2006 3rd IEEE Consumer Communications and Networking Conference*.

[98] Chaozhou Zhang, Min Li, Zonglin Wu, Chen-Guang Zhao, Hua Yuan, Jun Xie, Guanghua Xu, Jichun Li, and Shan Luo. 2024. A haptic feedback sleeve for a flight video game. *IEEE Transactions on Haptics* 18 (2024), 198–207.

[99] Zhenzhen Zhao, Nassim Laga, and Noël Crespi. 2009. A survey of user generated service. In *Proceedings of the 2009 IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 241–246.

[100] Samira Zomerplaag and Sander Bakkes. 2024. An exploratory study on gender dysphoria & character customisation. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, 1–12.

Received 7 July 2025; revised 26 November 2025; accepted 14 February 2026