User-Generated Content and Editors in Video Games: Survey and Vision

Haihan Duan, Yiwei Huang, Yifan Zhao, Zhen Huang, Wei Cai*
School of Science and Engineering, *The Chinese University of Hong Kong, Shenzhen*, China
{haihanduan, yiweihuang, yifanzhao, zhenhuang1}@link.cuhk.edu.cn, caiwei@cuhk.edu.cn

***Abstract*—User-generated content (UGC) is any form of content that has been created by users rather than the developers of online platforms. The UGC has been playing a very important role in video games. For instance, *Counter-Strike (CS)* and *Defense of the Ancients (DOTA)* originated from modifications of *Half-Life* and *Warcraft III: Reign of Chaos* respectively. As a promising trend, the UGC will be increasingly developed and extended to wider scopes in virtual worlds, such as the metaverse, which is highly promising for further study in both academia and industry. However, there are few existing surveys that systematically discuss the UGC in video games. In this paper, we systematically review the representative UGC in video games and their corresponding UGC editors based on a decision tree-style classification method. Then we enumerate the propagation methods of UGC in video games. Moreover, we propose our vision of the future development of UGC in the metaverse.**

***Index Terms*—User-Generated Content, Video Game, Game Editor, Metaverse.**

I. Introduction

The user-generated content (UGC) is any form of content that has been created by users rather than the developers of online platforms [1]. On the Internet, the classic UGC includes text, audio, video, animated material, etc [2], and these UGC represent the creativity, innovation, and delivery ability of users, which has shown an amazing power in promoting the development of online media communities [3].

Similarly, the evolution of video games is also benefit a lot from UGC [4]. The most famous cases are *Counter-Strike (CS)*[1] and *Defense of the Ancients (DOTA)*[2]. Thereinto, *CS* originated from a modification (MOD) of *Half-Life*[3], a first-person shooting (FPS) video game, while *DOTA* is a multiplayer online battle arena (MOBA) MOD of *Warcraft III: Reign of Chaos*[4]. The two representative games deeply improve the players' perspective, understanding, and acceptance of UGC in video games. Besides, the UGC in video games has various forms, including but not limited to game levels, user interface (UI), game character graphic alterations (so-called skins), etc [5], which significantly improves players' gaming experience. With the growing trend of UGC and their culture, community, and economics, the UGC gives the creators and players a sense of participatory involvement and also increases their enjoyment with digital games [6], [7].

*Wei Cai is the corresponding author (caiwei@cuhk.edu.cn).

[1]https://blog.counter-strike.net/

[2]https://www.dota2.com/home

[3]https://www.half-life.com/

[4]https://playwarcraft3.com/

There have been some related studies that review the development or impact of general UGC [2], [8]–[10]. Moreover, existing studies also conducted surveys on specific areas, including online brand communities [11], brands marketing [12], tourism [13], hotel experiences [14], digital well-being [15], social media [16], etc. However, there is only a small number of studies focusing on the UGC and their corresponding editors in digital games [4], [17], [18], while few of them illustrated detailed taxonomy or classification of the UGC in video games. Therefore, this paper intends to systematically survey the existing UGC and their corresponding content editors of some representative and notable video games, and provide our vision about the future development of UGC in the area of gamification, especially in the metaverse.

The remainder of this paper is organized as follows. We briefly review some previous works about UGC in video games in Section II. Then, we discuss and classify the UGC in video games in Section III and their corresponding UGC editors in Section IV. Afterward, we enumerate some representative propagation methods of video game UGC in Section V. And we propose our vision for the future development of UGC in video games in Section VI. Section VII concludes the paper.

II. Related Work

In recent years, the UGC in video games gradually prospered and had a unique and significant impact on the gaming field, which also attracted the attention of the academic community. For example, early in 2007, Lastowka *et al.* [17] studied the relationship between the UGC and general virtual worlds, in which the paper mainly discussed the practical and legal problems of the creation of UGC in virtual worlds. Kasapakis *et al.* [6] studied the UGC in a pervasive role-playing game for investigating the effect of UGC under diverse technical, functional, and gameplay characteristics. In 2015, Linberg *et al.* [4] conducted a quantitative study on members of 4 online video game UGC communities, and then deeply examined the effects of motivational factors of intrinsic motivation, extrinsic motivation, and toolkits that motivate game players to participate in UGC creation. Sotamaa *et al.* [5] mainly explained the concept, history, and types of game MODs, and also discussed the motivations of the creators and reactions of the game industry. However, the existing surveys did not illustrate a systematic review of the UGC and its creation tools, while they mainly focused on the social effects

of UGC in video games, such as the motivational factors of UGC creation, the perspective of the game industry, etc.

Correspondingly, the development toolkits or editors are necessary for creating the UGC and directly affect the user's creation experience. Currently, there are also some researchers who have studied how game editors influence game UGC creation using different technologies. Liapis *et al.* [19] introduced Sentient Sketchbook, a tool that supports a designer in the creation of game levels, which used map sketches to alleviate designers' efforts. Ng *et al.* [20] focused on the augmented reality (AR) environment and explored the concept and design of level editors in AR games. For studying the co-creation in video games, Davidovici-Nora [21] pointed out a new hybrid innovation model used to develop *World of Warcraft (WoW)*, a massively multiplayer online role-playing game (MMORPG). Besides, there are also many studies that applied artificial intelligence (AI) methods to support UGC creation in video games. For example, Summerville *et al.* [22] examined the use of Long Short-Term Memory (LSTM) recurrent neural networks for generating levels for *Super Mario Bros*. Guzdial *et al.* [23] presented an unsupervised process to generate full video game levels trained on gameplay video of *Super Mario Bros*. Moreover, a detailed discussion about how the design of an AI-driven game level editor affects creators was conducted by Guzdial *et al.* [24]. However, the published works mainly proposed novel methodologies or practical designs for the development toolkits and editors, while few of them systematically survey the existing game editors.

III. Categories of User-Generated Content

In video games, the imagination, innovation, and creative ability of players are unlimited, in which there are various UGC created with countless forms and styles, which makes the classification of the UGC using the traditional taxonomy methods difficult. Therefore, in this paper, we construct a decision tree-style classification Venn diagram according to the impact of the UGC on the basic video games, as shown in Fig. 1, and we also list some representative games for each type. Besides, Fig. 1 also contains the classification map of UGC editors, which will be studied in Sec. IV. Note that, one representative game will only be selected once although they might have multiple kinds of UGC and editors. In the following subsections, we will discuss the definition, characteristics, and representative games of each type in detail.

A. User-generated Game Modes

Currently, most games provide the default rules, maps, characters, and targets so that the players can follow the guidance to finish the whole story of the game. However, using the game as a framework, users can generate new game modes by modifying the original rules of the base games, which are user-generated game modes.

For the user-generated game modes, the representative point is that this kind of UGC provides new ways for gameplay. Or, the goal of the base games is to build a platform for users to create new modes. In this paper, the definition of game modes is not limited as discussed by Heintz *et al.* [25], while it has a more general scope for even tiny modifications. For example, in FPS games, if the user-generated game mode provides a shorter time to kill (TTK), it can be regarded as a new game mode. In fact, the shorter TTK mode already exists in many FPS games (e.g. *CrossFire*[5]). Therefore, we consider that the new game modes do not have to be completely different from the base games, but they should provide novel gaming experiences that are different from the basic rules or targets.

Generally speaking, we classify the user-generated game modes into four directions, including the change of game tempo, the change of game target, the change of game operation, and original game creation. The following part will discuss the representative prototypes of each direction. (1) The change of game tempo: A typical example is *Project Reality*[6] from *Battlefield 2*[7], in which the production team spent 17 years providing content-rich tactical shooters, and the shortening of TTK provides a different fight experience to the players. (2) The change of game target: The most notable case is the aforementioned *CS*[1]. In *CS*, there are two teams that have different goals on a closed map, where the terrorists need to plant and detonate a bomb while the Counter-strikers need to stop the terrorists. This user-generated game mode is totally different from *Half-Life*'s original team deathmatch[8] mode. (3) The change of game operation: *DOTA*[2] mentioned above is a representative example. Originally, the *Warcraft III*[3] is a real-time strategy (RTS) game, while *DOTA* changed the mode of controlling an army to only controlling a hero character. Besides, these directions can also be integrated, as the winning goal of *DOTA* is also different from *Warcraft III*, which can be regarded as the change of game target. (4) Original game creation: This category is somewhat special compared with the previous game MOD, in which a classic example is Roblox[9]. These games only provide basic rules, environments, materials, and operations, and players can create new game modes based on the provided resources, which contains the change of both game tempo, game target, and game operation.

B. User-generated Campaigns/Levels

Generally, the developers of games will set some targets or stories for the players to follow. After being familiar with the games, players may not satisfied with the original storyline and create new levels, new campaigns, or new explorable content. This kind of UGC possibly introduces new art, music, map, or model resources, but they will not change the original game mode and basic rules. This is a very common type of UGC in video games, especially for the games with map editors.

Regarding to the user-generated campaigns/levels, first of all, we consider they should be complete as campaigns/levels, which means they may have an independent portal rather than only a small piece of object in the base game. For example,

[5]https://www.smilegate.com/en/game/crossfire.do
[6]https://www.realitymod.com/
[7]https://www.ea.com/games/battlefield/battlefield-2
[8]https://en.wikipedia.org/wiki/Deathmatch
[9]https://www.roblox.com/

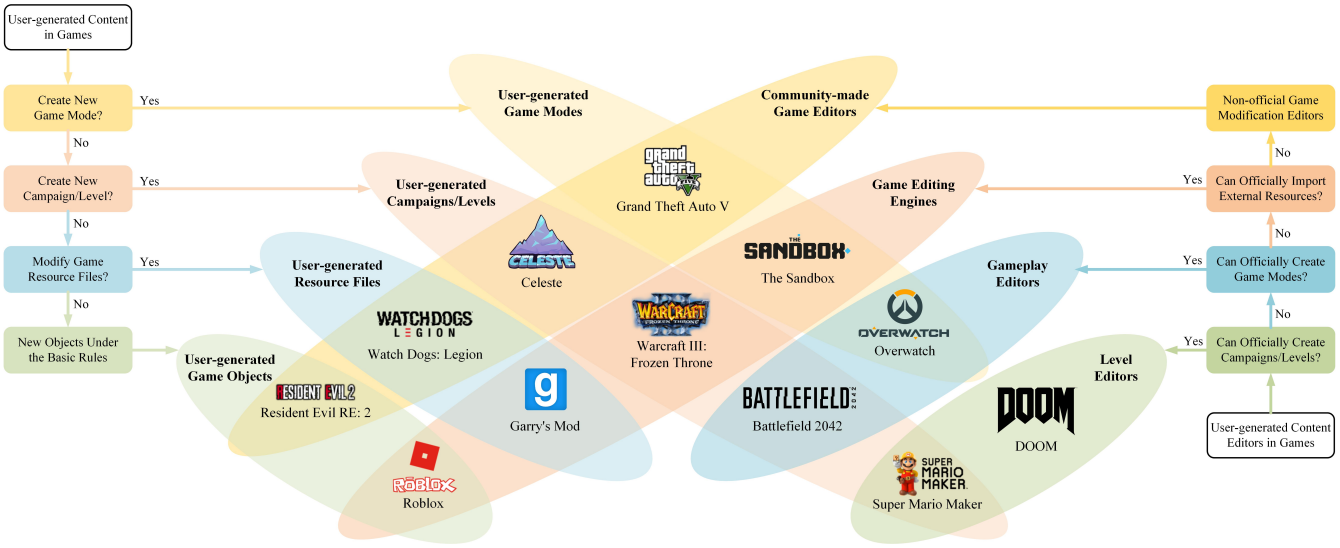

Fig. 1. Decision tree-like classification Venn diagram of user-generated content and the corresponding editors in video games.

in *Minecraft*[10], if the players created a building and moved it into their archive, it will not be counted as a user-generated campaign/level since it's too small and doesn't have a separate portal. But, if the players generated a whole map around the building with a corresponding storyline and game target, it will be regarded as a user-generated campaign/level, since other players could load the map individually through the game's level selection and explore the map. Second, the user-generated campaigns/levels should not change the game mode of the base game itself, such as a MOD of *Hearts of Iron IV*[11] named *The New Order: Last Days of Europe (TNO)*[12], which built a completely different world and history from the original game, and added a lot of external art and music resources, but *TNO* did not change the basic game mode of the original *Hearts of Iron IV*. Therefore, we still regard the *TNO* as a user-generated campaign/level of *Hearts of Iron IV*.

According to our survey, we find that the user-generated campaigns/levels mainly come from two sources: (1) Build-in map editor: There are many games that provide build-in map editors for players to customize the original environment. For example, *Age of Empires II: The Conquerors*[13], *Super Mario Maker*[14], *Forza Horizon 5*[15], etc, provide in-game map editors for players to create new levels. (2) External game resource modification: Some games equip with external game editors, which could modify outside resources to change the campaigns/levels, also known as modification (MOD). The aforementioned *TNO* and *The Elder Scrolls V: Skyrim*'s[16] story MOD *The Forgotten City*[17] are both outside MODs. More famously, the notable *CS* map *DUST2* also belongs to user-generated campaigns/levels, which started as a map MOD that can be downloaded from forums and then was officially incorporated into the game due to its great reputation.

C. User-generated Resource Files

In some games, the users need to directly modify the game files or resources to create UGC. The well-known MODs in video game community refer to these user-generated resource files and feature new files imported from outside of the game. Common forms of user-generated resource files include modifying art resources, audio resources, action resources, new items, new characters, new weapons, and so on. It is worth mentioning that if the added resource is a separate map, we will regard it as a user-generated campaign/level, which is discussed in Sec. III-B. Therefore, in our consideration, the user-generated resource files are usually small items that need to modify the original resources to achieve the players' goal. For example, there is a suit of clothing for engineers in the original game, but the user privately change it to a suit for fishing by modifying the resource file, which would be regarded as the user-generated resource files.

Since the user-generated resource files are MODs imported from outside of the game, there is basically no official production tool. On the other hand, creating these UGC needs to analyze the source codes of the game, which is more common in open-source games such as *Half-life*[3]. Therefore, the threshold of producers for creating the user-generated resource files is relatively high, since they need to have the ability of programming. However, although there are not many creators that have these abilities and skills, the players who are really willing to make this kind of UGC are basically the core players who love the game deeply. Generally, these players are more likely to produce high-quality UGC, which is highly beneficial for the development of UGC in the game. However, with the game developers increasingly focusing on official

[10]https://www.minecraft.net/
[11]https://www.paradoxinteractive.com/games/hearts-of-iron-iv/about
[12]https://steamcommunity.com/sharedfiles/filedetails/?id=2438003901
[13]https://www.ageofempires.com/
[14]https://supermariomaker.nintendo.com/
[15]https://www.xbox.com/en-US/games/forza-horizon-5
[16]https://elderscrolls.bethesda.net/en/skyrim
[17]https://forgottencitygame.com/

editors, the number of creators struggling for user-generated resource files is decreasing compared with other categories.

As mentioned above, the production of the user-generated resource files needs to analyze and modify the original game files, so not all game companies are willing to allow players to do so. For example, *Rockstar Games*[18], the maker of the notable *Grand Theft Auto* series, seems an opponent of the MODs in their published games. Although the MODs of *Grand Theft Auto V* are really abundant and impressive, including various vehicles, characters, etc, *Rockstar Games* explicitly prohibit the modification of *Grand Theft Auto V*[19] in their End User License Agreement[20]. On the contrary, *VALVE Corporation*[21], the maker of *Half-life* series, actively discloses the source code to encourage creators to make new MODs for their games. Moreover, due to the popularity of *CS*, *VALVE* even acquired the production team of *CS* and supported them to develop *CS* independently which became famous video games around the world, and so did *Day of Defeat*[22]. In this paper, we do not intend to compare the solution and attitude of game companies toward the user-generated resource files, since some MODs may break the balance of competition in multi-player games (so-called plug-in) or even violate policies and cause legal issues, which would be harmful for the games and companies. In most cases, the useful and interesting game MODs are very important and show a highly positive impact on the game community, which also fosters many good game producers who were MOD creators before their formal work.

D. User-generated Game Objects

The user-generated game objects are most common UGC in video games. The players can use the provided game editors to create the UGC of game objects without importing outside resources. The creation of the user-generated game objects has the simplest threshold compared with other categories, since most of the creation approaches are bound to the games' own play mode, so the players do not need to learn extra operation and logic. This is the most intuitive way for normal users to feel the sense of achievement when creating UGC, which activates the players' innovation ability and also extends the games' life cycle. But correspondingly, this creation approach shows great limitations and low degree of freedom during the editing. For example, many games only allow players to customize the color scheme of characters, while not truly allow them to change their models, actions, or audio, etc.

The user-generated game objects exist in many notable games. For example, in *Escape from Tarkov*[23], players can modify their weapons as they wish. In *Besige*[24], players can freely construct their favorite machines to bit against castles or armies. Sports games also allow the users to customize team members, e.g. *Pro Evolution Soccer*[25] allows users to create new players by setting their ratings, faces, actions, etc. More simply, the players can just paint the characters to shows their personality in *Human: Fall Flat*[26].

On the other hand, the sandbox games are special cases. Many sandbox games or physical simulation games (e.g., *Minecraft*[8], *The Sandbox*[27], *Cryptovoxels*[28], etc) do not set clear game targets, while they provide platforms for the players to create UGC. Although these sandbox games have a powerful degree of freedom and creation potential, most common players might only use them to create some small customized items, like a house, a tree, a piece of furniture, etc. Therefore, we regard this kind of UGC as user-generated game objects, while the huge and elegant scenes will be classified as user-generated campaigns/levels discussed in Sec. III-B.

IV. Categories of User-Generated Content Editor

In the previous section, we discuss the classification of the UGC in video games. Correspondingly, we also build a decision tree-style classification Venn diagram for UGC editors as shown in Fig. 1, which contains four types, including level editors, gameplay editors, game editing engines, and community-made game editors. The details of the UGC editors will be discussed in the following subsections.

A. Level Editors

To enhance the playability and longevity of video games, many existing products allow users to customize campaigns/levels. This kind of UGC editor usually does not allow users to create new rules that are different from the base game, while only allowing players to use existing elements to make new levels based on the basic rules of the games. With level editors, players can only expand the gaming experience rather than enhance the experience or create a new experience, and it is a great way for the players to extend the lifespan of a game by themselves. Majority of the official game UGC editors falls into the level editors, especially in the early days of video games, e.g. console-based games like the *Nintendo Entertainment System (NES)*[29] and *Sega Genesis*[30], in which the players can only modify or optimize a small part of the campaigns/levels due to the limitation of the hardware.

Currently, many modern games make it much easier to implement the level editors instead of other more complex editors that can change the game rules, so most existing notable video games officially provide level editors with the base games. For example, the editor in the early *NES* game *Excite Bike* only allows players to re-permutate the pre-made obstacles to create new levels, which is a small extension based on the original game without breaking any existing rules. Specifically, level editors have enormous popularity in RTS

[18]https://www.rockstargames.com/
[19]https://www.rockstargames.com/gta-v
[20]https://www.rockstargames.com/eula?country=us
[21]https://www.valvesoftware.com/
[22]https://www.dayofdefeat.com/
[23]https://www.escapefromtarkov.com/?lang=en
[24]https://www.spiderlinggames.com/
[25]https://www.konami.com/wepes/2021/us/en-us/ps4/
[26]https://www.nobrakesgames.com/humanfallflat
[27]https://www.sandbox.game/en/
[28]https://www.cryptovoxels.com/
[29]https://www.nintendo.com/nes-classic/
[30]https://genesismini.sega.com/

games, like *Warcraft III: Frozen Throne*, *StarCraft II: Wings of Liberty*[31], *Age of Empires II: The Conquerors*, etc. Moreover, many FPS games also provide level editors and contribute to sustain the improvement of the games, e.g. *DOOM*[32] and *Halo3*[33]. Specifically, we can find that the players' reactions to these UGC in *DOOM* are highly positive, since the MOD communities of *DOOM* are still active even thought *DOOM* was developed about 30 years ago, such as *MODDB*[34].

B. Gameplay Editors

Gameplay editors denote the UGC editors that can create new rules (or break existing rules) different from the original games. The gameplay editors not only allow players to use existing elements in the games' levels but also to change the rules to make new content. In this paper, we name this kind of editor as gameplay editors since the created UGC is possibly not limited to the original game, so the created UGC can completely abandon the original rules and even make a new game based on the basic elements. With gameplay editors, players can have more freedom than level editors, which may be helpful for building a long-lasting creator community and expanding the longevity of the games with endless creativity.

Gameplay editors are getting more and more popular today. For example, the most recently renowned one is the editor in the AAA (3A) game titled *Battlefield 2042*[35]. With this gameplay editor, players managed to create a "squid game" mode based on the famous Korean TV play, which represents the high potential and freedom of gameplay editors. Moreover, in the gameplay editor of *Overwatch*[36], an FPS game, the players can use the resources provided by the gameplay editor and its rich customization tools to create games in completely different genres such as a 2D platform game (often simplified as platformer or jump and run games, e.g. the early version of *Super Mario Bros*), hide-and-seek, zombie mode, etc.

By comparison, the high potential of gameplay editors also comes with high development costs due to the complexity of creating those editors. Basically, the development workload almost equals to developing an in-game editing engine, which explains why gameplay editors are more commonly seen in 3A titled games instead of indie titled games. On the other hand, the gameplay editors also come with the limitation of the original game's resources. Generally, most official gameplay editors will provide ways for creators to spread their UGC to other players in the game. So, in an official game editor, players usually cannot import their external resources into the game due to the developer's consideration of the legibility and potential copyright issue of the external resources and files.

C. Game Editing Engines

Different from the gameplay editors, game editing engines have richer features compared with the gameplay editors, like real game engines. Specifically, we consider the main difference that separate game editing engines and gameplay editors is the external resource. Game editing engines can bring assets from outside of the game to make brand new content that look completely different from the base games, while the gameplay editors only allow creators to use assets within the game. Sometime, those UGC can be new game genres or even be made as a standalone game. Both the Battle Royale [26] mode and MOBA are started as UGCs created in such game editing engines. For example, *DOTA* was created using *Warcraft III*'s game editing engine, which was further developed as a standalone game separate from the base game.

Another difference between game editing engines and gameplay editors is that the content created in game editing engines are not officially distributed in the game by the developer due to the copyright concern from the external assets imported by the creators. Thus, the UGC created by game editing engines are normally distributed by the creators themselves as the form of MODs. For instance, some variations of *DOTA* are distributed by the player community, and the players can play them using the official *Battle.net*[37] server or using the local area network (LAN) to connect with each other.

Specifically, a small quantity of UGC is made using game editing engines but also is distributed by the developers. For example, in *Roblox Studio*[38], a game editing engine of *Roblox*, players can make their custom assets. Even though *Roblox Studio* basically does not provide base game content, the strong capability of its game editing engine gives *Roblox* more and more UGC over the last 10 years. But, it is worth mentioning that the UGC created by *Roblox Studio* will be censored before it can be distributed to the players, and the strict copyright system guarantees the quality of UGC in *Roblox*. Correspondingly, the system of *Roblox* also costs a certain degree of the workload for its developers and operators. Though a game editing engine is much harder to construct and maintain than a game editor or a level editor, the amount and quality of the UGC can also achieve a higher level. According to the experience of *Roblox Studio*, we consider the game editing engine may be the most ideal tool for game UGC.

D. Community-made Game Editors

As discussed in Sec. III-C, many game companies do not provide official game editors because of many different reasons, e.g. legal considerations, corporate culture, etc. Therefore, many professional players in game communities would develop non-official game editors for modifying and creating content of the games, which are called community-made game editors in this paper. The core difference between community-made game editors and official game editors is the maintenance, which means the community-made game editors are created and maintained by the player community without the participation of the official game developers.

Most community-made game editors change the games through integrating MODs to the original games, including

[31] https://starcraft2.com/
[32] https://bethesda.net/en/game/doom
[33] https://www.halowaypoint.com/en
[34] https://www.moddb.com/games/doom/mods
[35] https://www.ea.com/games/battlefield/battlefield-2042
[36] https://playoverwatch.com/en-us/
[37] https://www.blizzard.com/en-us/apps/battle.net/desktop
[38] https://www.roblox.com/create

but not limited to changing the player states, modifying game objects' models, adding new maps, creating customized game modes, etc. The community-made game editors mainly rely on modifying game files by the players, which are mostly available on personal computers (PCs) rather than consoles. Although there are MODs and corresponding editors on hacked versions of consoles, these editors often have higher threshold. Thus, compared with the previous three kinds of editors, community-made game editors are only designed for a small proportion of enthusiasts to customize their favorite games. More importantly, it is common that only the popular games can have a dedicated community to make various kind of editors, and the abundant editors also play an important role in extending the game life-span, which builds a positive circulation for the popular games at the same time.

The most common kind of community-made game editors that can be seen in most PC game is trainers, which could modify memory values of games based on memory addresses. For example, a community-made editor of *Pro Evolution Soccer* allows users to adjust the player ranking, club transferring, and player value, etc. Another kind of popular community-made game editors is model swapper, which allows players to swap the in-game models of game objects (e.g. changing the model of the main character). For instance, in *Grand Theft Auto V*, players can use swapper to change the model of characters, passers-by, vehicles, etc. Moreover, there are some community-made game editors allow players to have full control of the camera and adjust its angle to have a better view of the game world and even record cinematic videos. For example, players made a plug-in for *Resident Evil 2*[39] to change the field of view (FOV) of camera, introduce manual flashlight, and allow players to swap characters/outfits.

V. Propagation of User-Generated Content

To build a positive circulation of UGC in video games, the propagation of UGC plays a necessary role through lots of approaches, like player community sharing, circulation in trading markets, combination with original games, etc. The following subsections will discuss these methods.

A. Player Community Sharing

Sharing in the player community is a common way to spread the game UGC, which we consider is the most widespread and fast way for UGC propagation. For the area of video game, the player community mostly refers to the online forum and chatting group for players to share their gaming experience, gaming strategies, etc. The motivation of player community sharing mainly comes from two perspective: (1) It comes from the true affection of the base games. The sharing of high-quality UGC in the player community may raise a certain degree of attention and attract more players to join the game, which could motivate more creation of UGC based on the popular one, and the participation of more players also drives the game companies to constantly maintain the game. (2) It comes from the popularity and competition of the creators in UGC communities. The spread of UGC could meet the needs of UGC creators for sharing their ideas and creations with other players, enriching the emotional experience of creators, and generating a certain incentive for the UGC creators. On the other hand, the player community is the most intuitive way for communicating with the UGC creators, which could build a positive feedback about the UGC and help their improvement.

There are many notable examples of UGC that were famous for the sharing in the player community. For example, a notable *Warcraft III* battle map named *Chenghai 3C*[40] enjoyed equal popularity with *DOTA* in early years, which was uploaded on online forum and then propagated in chatting groups and LAN. Besides, in lots of large online game communities like *Modyolo*[41] and *apkmody*[42], the players would freely upload their created MODs, patch resources, and even self-made editors for other players to use and share.

B. Circulation in Trading Markets

In some games, a part of the UGC mainly focuses on building a unique appearance or special usability, which has a strong appeal to some players, so the circulation in the trading market becomes one of the main ways for this UGC propagation. UGC circulation in the trading market means that UGC creators can trade their creations on online platforms for economic consideration, and players would buy the UGC according to their preference for the UGC's practical value, collection value, reputation, etc. Therefore, the two sides in the trading market can build the UGC transactions to achieve the ultimate effect of mutual benefit and a win-win. One of the trading methods is on the game's officially available UGC trading platform (e.g. *The Sandbox Shop*[43]), where UGC creators and the game players can trade the UGC certified on the official platform. Another way to trade UGC between users is peer-to-peer (P2P) transactions, where UGC creators and buyers can connect each other using third-party platforms.

Representative examples of circulation in the marketplace include *The Sandbox Shop*, *Cryptovoxels Marketplace*[44], *Steam Workshop*[45], etc. In *The Sandbox Shop*, players can trade user-generated entities, equipment, wearables, avatars, etc. Similarly, wearables, voxel artworks, and customized objects can be built in *Cryptovoxel* and traded in *Cryptovoxels Marketplace*. More importantly, these transactions will be confirmed on blockchain to guarantee the ownership of the digital assets, which will be further discussed in Sec. VI. Moreover, *Steam Workshop* may be the most common UGC trading market, which is a central hub of player-created content and tools to publish, organize, download, and trade UGC for corresponding games. Specifically, for the games created by *VALVE* (e.g. *Counter-Strike: Global Offensive (CS: GO)*[1]),

[39] https://www.residentevil2.com/us/

[40] http://www.ch3c.cn/

[41] https://modyolo.com/

[42] https://apkmody.io/

[43] https://www.sandbox.game/en/shop/

[44] https://www.cryptovoxels.com/marketplace

[45] https://steamcommunity.com/workshop/

Steam Workshop would encourage the players to create UGC and publish them freely, and *VALVE* will share out bonuses according to the popularity and voting of these UGC.

C. Combination with the Original Games

Through the official UGC creation platform, players can create various UGC with their imagination and creativity, like unique game levels with higher playability and different gaming experience. For outstanding UGC, the official developers may integrate these UGC with the original game to extend them as a part of the game. In this way, UGC can be used and appreciated by more players, which not only efficiently utilizes the innovation of the UGC creators but also increases the richness and diversity of the game content and constantly attract the players of the game. From the perspective of creators, the official certification can also motivate them to build high-quality UGC. Therefore, the combination of UGC with the original games is also a promising method for UGC propagation and development in video games.

For the combination of UGC and original games, *VALVE* can be a representative leader. Specifically, the most famous case is the integration of *CS* and player-made maps. In the early version of *CS*, the map *DUST2* was a user-generated map that was famous around online communities. In the subsequent releases, the map *DUST2* was officially integrated into *CS* and also spread to other FPS games later, which makes it highly notable that almost all FPS players are familiar with the map *DUST2*, even though they might not be original players of *CS*. Another example is *Team Fortress 2*[46], a cartoon-like FPS video game developed by *VALVE*. In 2013, *Team Fortress 2* added the first community-made update, named *Robotic Boogaloo*[47], which contained 57 new items including in-game content, the update hub website, the animated short, the comic, and the splash images in the Steam store.

VI. Vision of User-Generated Content

With the development of network capacity, physical engine, and computational power, the video game will show a great improvement beyond the scope of entertainment, and we believe the final form of the video game is **metaverse**. Metaverse denotes the next-generation Internet in which users can control their avatars to interact with other users and software applications in a 3D virtual space [27]. Different from the common video games, in metaverse, there are no specific targets for players to chase, while the players can freely play for entertainment, creation, and even work. Theoretically, the metaverse has unlimited extension space for UGC creation and seamless scene transformation, so the construction of metaverse will highly depend on the users' creativity and innovation [27]. Therefore, we believe the creation, artistic style, behavior, culture, and economy of UGC in metaverse will be significantly promising research areas, and we conclude two representative directions in the following subsections.

[46]https://www.teamfortress.com/

[47]https://wiki.teamfortress.com/wiki/Robotic_Boogaloo

A. Evolution of User-Generated Content Editors

For promoting the UGC creation, the most intuitive requirement is an efficient UGC editor/tool that could facilitate the creation procedure and lower the threshold for all new players. Although there is multiple existing 3D modeling software (like *Maya*[48], *Blender*[49]), and UGC editors discussed in Sec. IV, most existing editors are highly dependent on professional knowledge and practical experience, so it is difficult for new players to use. Therefore, it is imperative to make breakthroughs for the evolution of UGC editors/tools.

In recent years, AI-related technologies have shown great improvements, especially in content generation (e.g., generative adversarial network (GAN) [28]). With the help of GAN, it is easier for new players to create beautiful pictures with handy sketches [29] and fine-grained models by low-resolution models [30], so the integration of game UGC editors and GAN is a potential trend for improving the UGC editors. Moreover, procedural content generation (PCG) can also be a helpful solution [31]. The UGC editors can provide PCG-based modules to help creators to construct the virtual environment (e.g., buildings, trees, grasses, plants, mountains, planets, etc).

Specifically, we also need to point out the impact of artistic style in UGC creation, especially voxels. The voxel style has shown a great potential for UGC creation in many popular games (e.g., *Minecraft*, *Cryptovoxels*, *Lego Worlds*[50], etc). Compared with other artistic styles, the voxel style has its special advantages: (1) Voxel style objects are quite simple and intuitive for UGC creation and modification with low learning cost; (2) Most voxel style avatars are abstract, so the uncanny valley problem may be effectively avoided; (3) The voxel style modeling can unlock new simulation techniques (e.g. simulation of smart materials [32]), which is hard for other modeling methods. Therefore, in our vision, we consider the voxel style may be the mainstream in future metaverse applications, and the corresponding problems of voxel modeling like the physical engine (e.g. collision, fragmentation, etc) and AI-assisted UGC generation will be promising research topics.

B. Ownership and Economy of User-Generated Content

The economic system is necessary for a large-scale virtual world, also known as a virtual economy [33]. With the development of blockchain-related technologies, the decentralized finance (DeFi) with the digital currency of ERC-20 Token Standard[51] can be naturally helpful for building the economic system of the metaverse. Moreover, ERC-721 Non-Fungible Token (NFT) Standard[52] can certify the digital assets as unique and not interchangeable tokens on the blockchain, which means the UGC in video games could truly belong to the users while not depend on the operating companies of the games. Under this circumstance, the UGC in video games can truly have specific values as NFTs rather than only represent

[48]https://www.autodesk.com/products/maya/overview

[49]https://www.blender.org/

[50]https://www.lego.com/en-gb/categories/games/lego-worlds

[51]https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[52]https://eips.ethereum.org/EIPS/eip-721

abstract feelings or enthusiasm for the game, and the values of the UGC may depend on multiple factors, such as their quality, scale, rarity, iconic meaning, creators' fame, etc.

Moreover, the online markets of both official marketplace (e.g. *Cryptovoxels*) and third-party exchange (e.g. *Opensea*[53]) will provide liquidity for UGC. On these platforms, players can freely select their preferred digital assets, and creators can build objects with higher quality to earn higher rewards. Thus, a creator economy ecosystem with positive circulation is built, in which the players will have higher user stickiness due to their favorite UGC and the corresponding cost, and the notable creators also become the builders of the metaverse who can earn for life like the architects in the real world. And this exciting UGC creator economy will attract more participants of both creators and players, thus enhancing the ecosystem again. Therefore, the study of the blockchain-based creator economy is necessary for the further improvement of the UGC.

VII. Conclusion

In recent years, UGC has gradually become an important feature and component of outstanding video games, and in the future, UGC will play a more necessary role in building the metaverse. This paper reviews and classifies the existing UGC and the corresponding editors, and lists the propagation approaches of UGC. More importantly, this paper proposes our vision for the development of UGC, which focuses on the promising prospects and research directions in the metaverse, including the evolution of UGC editors and the ownership and economy of UGC. We believe the prosperous game industry will benefit from the creativity and innovation of UGC, and the forms, values, and meanings of UGC will also achieve a higher level during the building of metaverse.

Acknowledgment

This work was supported by Shenzhen Science and Technology Program (Grant No. JCYJ20210324124205016).

References

[1] T. Min, H. Wang, Y. Guo, and W. Cai, "Blockchain games: A survey," in *2019 IEEE conference on games (CoG)*. IEEE, 2019, pp. 1–8.

[2] T. K. Naab and A. Sehl, "Studies of user-generated content: A systematic review," *Journalism*, vol. 18, no. 10, pp. 1256–1273, 2017.

[3] Z. Zhao, N. Laga, and N. Crespi, "A survey of user generated service," in *2009 IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 2009, pp. 241–246.

[4] J. Lundmark and E. Sandström Lindberg, "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," 2015.

[5] O. Sotamaa and H. Wirman, "Online games modifications and user generated content," *The International Encyclopedia of Digital Communication and Society*, pp. 1–10, 2015.

[6] V. Kasapakis and D. Gavalas, "User-generated content in pervasive games," *Computers in Entertainment (CIE)*, vol. 16, no. 1, pp. 1–23, 2017.

[7] R. L. S. Díaz, G. R. Ruiz, M. Bouzouita, and K. Coninx, "Building blocks for creating enjoyable games—a systematic literature review," *International Journal of Human-Computer Studies*, p. 102758, 2021.

[8] J. Krumm, N. Davies, and C. Narayanaswami, "User-generated content," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 10–11, 2008.

[9] C. Wyrwoll, "User-generated content," in *Social Media*. Springer, 2014, pp. 11–45.

[10] X. Ochoa and E. Duval, "Quantitative analysis of user-generated content on the web," 2008.

[11] A. Estrella-Ramón and F. Ellis-Chadwick, "Do different kinds of user-generated content in online brand communities really work?" *Online Information Review*, 2017.

[12] G. Christodoulides, C. Jevons, and J. Bonhomme, "Memo to marketers: Quantitative evidence for change—how user-generated content really affects brands," *Journal of advertising research*, vol. 52, no. 1, p. 53, 2012.

[13] D. C. Ukpabi and H. Karjaluoto, "What drives travelers' adoption of user-generated content? a literature review," *Tourism management perspectives*, vol. 28, pp. 251–273, 2018.

[14] A. Barreda and A. Bilgihan, "An analysis of user-generated content for hotel experiences," *Journal of Hospitality and Tourism Technology*, 2013.

[15] M. T. Cuomo, D. Tortora, A. Giordano, G. Festa, G. Metallo, and E. Martinelli, "User-generated content in the era of digital well-being: A netnographic analysis in a healthcare marketing context," *Psychology & Marketing*, vol. 37, no. 4, pp. 578–587, 2020.

[16] M. Luca, "User-generated content and social media," in *Handbook of media Economics*. Elsevier, 2015, vol. 1, pp. 563–592.

[17] G. Lastowka, "User-generated content and virtual worlds," *Vand. J. Ent. & Tech. L.*, vol. 10, p. 893, 2007.

[18] C. S. Brunt, A. S. King, and J. T. King, "The influence of user-generated content on video game demand," *Journal of Cultural Economics*, vol. 44, no. 1, pp. 35–56, 2020.

[19] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: computer-assisted game level authoring," 2013.

[20] G. Ng, J. G. Shin, A. Plopski, C. Sandor, and D. Saakes, "Situated game level editing in augmented reality," in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, 2018, pp. 409–418.

[21] M. Davidovici-Nora, "The dynamics of co-creation in the video game industry: the case of world of warcraft," *Communications & Strategies*, no. 73, p. 43, 2009.

[22] A. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," *arXiv preprint arXiv:1603.00930*, 2016.

[23] M. Guzdial and M. Riedl, "Game level generation from gameplay videos," in *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[24] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, and M. O. Riedl, "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*, 2019, pp. 1–13.

[25] S. Heintz and E. L.-C. Law, "The game genre map: A revised game classification," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, 2015, pp. 175–184.

[26] G. Choi and M. Kim, "Gameplay of battle royale game by rules and actions of play," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. IEEE, 2018, pp. 599–600.

[27] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 153–161.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[29] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425.

[30] Z. Chen, V. G. Kim, M. Fisher, N. Aigerman, H. Zhang, and S. Chaudhuri, "Decor-gan: 3d shape detailization by conditional refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 740–15 749.

[31] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," *Neural Computing and Applications*, vol. 33, no. 1, pp. 19–37, 2021.

[32] G. Sossou, F. Demoly, H. Belkebir, H. J. Qi, S. Gomes, and G. Montavon, "Design for 4d printing: Modeling and computation of smart materials distributions," *Materials & Design*, vol. 181, p. 108074, 2019.

[33] M. Nazir and C. S. M. Lui, "A brief history of virtual economy," *Journal for Virtual Worlds Research*, vol. 9, no. 1, 2016.

[53] https://opensea.io/