Asia Pacific Academy of Science Pte. Ltd.

Metaverse 2024, 5(2), 2904.
https://doi.org/10.54517/m2904

Review

Video game development 3.0: AI-driven collaborative co-creation

Jeremiah Ratican, James Hutson*

Lindenwood University, MO 63301, United States
*** Corresponding author:** James Hutson, JHutson@lindenwood.edu

CITATION

Ratican J, Hutson J. Video game development 3.0: AI-driven collaborative co-creation. Metaverse. 2024; 5(2): 2904. https://doi.org/10.54517/m2904

ARTICLE INFO

Received: 27 August 2024
Accepted: 20 November 2024
Available online: 17 December 2024

Abstract: The evolution of game development has transitioned from manual coding (Software 1.0) to data-driven Artificial Intelligence (AI) (Software 2.0), and now to a more advanced stage—video game development 3.0. This phase is characterized by AI-driven processes leveraging large language models (LLMs), neural networks, and other AI techniques that autonomously generate code, content, and narratives. This paper explores the foundational technologies underpinning this paradigm shift, including customizable AI modules, dynamic asset creation, and intelligent non player characters (NPCs) that adapt to player interactions. It also highlights the integration of AI with emerging technologies like Virtual Reality (VR), Augmented Reality (AR), and brain-computer interfaces, which are set to further enhance player immersion. Through case studies of AI-driven procedural world creation in No Man's Sky, dynamic narrative generation in RPGs like Cyberpunk 2077, and AI-assisted esports balancing in League of Legends, the study demonstrates the transformative potential of AI in creating more scalable, adaptive, and personalized game experiences. However, the paper also addresses challenges, such as balancing technical assistance with human creativity and ensuring ethical practices in AI-driven content. Ultimately, this study envisions a future where AI democratizes game development, empowering both indie developers and large studios to create more innovative and inclusive games. The conclusion calls for collaboration among developers, researchers, and players to fully realize the potential of the new generative framework, emphasizing the need for continuous exploration and refinement of AI tools in gaming.

Keywords: AI-driven game development; large language models; procedural content generation; intelligent NPCs; collaborative AI-human design

1. Introduction

The evolution of software development has significantly impacted the video game industry, transitioning from manual coding (Software 1.0) to data-driven AI methods (Software 2.0). In the early days, Software 1.0 primarily relied on manual programming, where every element of a game was painstakingly coded by developers [1]. This era was characterized by rigid, predefined behaviors in games, which limited adaptability and scalability [2]. As games grew in complexity, the need for more dynamic, data-driven approaches led to the rise of Software 2.0. This phase introduced machine learning and other AI techniques, allowing games to incorporate adaptive behaviors, such as non-playable characters (NPCs) that could learn from player actions and environments that could dynamically adjust based on in-game conditions [3,4].

Machine learning (ML) has become a cornerstone of modern game development, enabling features like procedural content generation and adaptive AI in games [5]. For instance, titles like No Man's Sky (2016) utilize AI to generate vast, ever-changing environments, offering a level of diversity and complexity that would be impractical

to achieve through manual coding alone [6]. Additionally, reinforcement learning and genetic algorithms have been employed to create NPCs capable of evolving strategies and behaviors based on player interaction, contributing to more immersive and engaging gaming experiences [7]. These advancements have laid the groundwork for Software 3.0, which seeks to push these innovations further by leveraging neural networks and large language models (LLMs) to autonomously generate not just code, but also entire game assets and narratives [8]. Unlike the more traditional Software 2.0, which focused on ML models driven by predefined tasks and structured datasets, Software 3.0 harnesses the broader capabilities of LLMs such as GPT-3, Codex, and other advanced generative models [9]. These models are capable of interpreting high-level instructions, allowing them to produce detailed code, art, and narrative elements autonomously, thereby facilitating a more dynamic and creative development process [10]. OpenAI's Codex, for example, is a pioneering tool in this domain, able to translate natural language prompts directly into functional code, thereby reducing the manual workload traditionally associated with software development [11].

One of the central advantages of Software 3.0 is its procedural content creation capabilities. Through the integration of neural networks, game assets, environments, and even complex storylines can be generated with minimal human intervention. LLMs contribute by learning from vast amounts of data, enabling them to produce contextually relevant and diverse content [12]. As Nasir et al. [13] note, this not only accelerates the development timeline but also enhances creativity by allowing developers to iterate more rapidly on ideas without being constrained by traditional design bottlenecks. The transition towards autonomous systems is expected to democratize game development further, enabling indie developers to create content with the same complexity and polish as larger studios by relying on AI-powered tools [14].

The thesis of this paper thus centers on the emergence of new strategies for video game development, which represents the next evolutionary step in game development through AI-driven processes. As the complexity and scale of video games continue to expand, traditional development methods increasingly fall short. The rise of Software 3.0 in the broader technology landscape—marked by neural networks and AI systems capable of autonomous learning and decision-making—has begun to transform software engineering and development practices across multiple industries, including gaming [15]. In video game development 3.0, AI is not merely a tool but a driving force behind co-creative processes such as procedural content generation, dynamic storytelling, and adaptive game mechanics [16]. This shift allows for the automation of traditionally human-centered tasks like level design and narrative crafting, which in turn reduces development time and scales creativity in unprecedented ways. Therefore, approaching game development from this AI-centric perspective introduces new possibilities for the industry. Autonomous systems can analyze large datasets of player behavior, generate personalized content, and even refine in-game mechanics in real-time to match evolving player preferences [17]. The results indicate that the proposed framework can significantly improve both the efficiency of game production and the quality of the player experience by integrating these tools directly into the core development process. The significance of this transformation lies in its potential to

redefine the roles of developers and designers, enabling them to focus on higher-level creative decisions while AI handles the technical complexities [2].

2. Background

Traditional game development processes have historically been resource-intensive and laborious, relying heavily on manual asset creation, static scripts, and hard-coded behaviors for NPCs (**Figure 1**). In this context, developers often needed to code each element of the game individually, including graphics, sound, and behavior scripts, which demanded extensive labor from artists, writers, and programmers [18]. The limitations of these methods are evident in the repetitive and simplistic behavior of NPCs, which typically relied on predefined rules or finite state machines that offered little variation or adaptability [19]. Furthermore, creating immersive and interactive game worlds required developers to manually script behaviors for hundreds, if not thousands, of NPCs, a process that was time-consuming and challenging to scale as games became more complex [20]. These constraints have historically limited the richness of game narratives and interactions, highlighting the need for more scalable and dynamic solutions.

Figure 1. Goitch by wuzzy (minetest) (CC 4.0).

Figure 1 depicts a procedurally generated game environment created using Minetest, an open-source voxel game engine. This image illustrates the capabilities of AI-driven procedural content generation, where game environments are built automatically using predefined algorithms and AI models. In this context, "procedural content generation" refers to the automation of world-building processes, significantly reducing the manual labor traditionally required for game design. The figure emphasizes the scalability of these processes, allowing developers to create vast, diverse game worlds with minimal input. This procedural approach exemplifies how AI can rapidly generate environments that maintain aesthetic consistency while offering enough variation to keep gameplay engaging.

In response to the inefficiencies of traditional development processes, AI-driven

methods have become increasingly integrated into game development, laying the foundation for what could be termed as video game development 3.0. One of the most prominent applications of AI is in procedural content generation, where algorithms autonomously create game environments, levels, and assets. Games like No Man's Sky (2016) exemplify this approach, offering vast, procedurally generated worlds that evolve dynamically based on predefined parameters [16]. Additionally, AI has been used to implement adaptive difficulty systems that adjust gameplay based on player behavior and skill levels. For instance, in Left 4 Dead (2008), the AI Director tailors the intensity of gameplay in real-time, making each session unique and challenging [21]. These innovations not only enhance player experience but also significantly reduce the manual workload traditionally required in game design, thereby streamlining development and enabling more creative flexibility.

The increasing complexity of modern video games and the growing expectations of players have revealed significant limitations in traditional development methods. Game worlds are becoming larger and more intricate, with a higher demand for dynamic content and adaptive gameplay that responds in real-time to player interactions. The increased complexity requires more scalable and intelligent development processes, where manual scripting and static content are no longer sufficient to meet the industry's needs [22]. Current development cycles are often characterized by extensive coding, iterative design processes, and high resource demands, which hinder both creativity and efficiency [2].

To address these challenges, the gaming industry requires a new paradigm that leverages the full potential of intelligent systems to automate and enhance creative processes. AI-driven systems have already shown significant promise in procedural content generation, enabling the creation of complex environments and narratives with minimal human input. These systems not only streamline development but also allow for the dynamic generation of game elements that evolve based on player behavior and game context [23]. Moreover, this smart technology has the ability to analyze vast amounts of player data and adapt gameplay accordingly ensuring that experiences remain engaging and personalized, contributing to better retention and satisfaction rates. The shift towards AI-centric development practices represents more than just technological progress; it is a fundamental rethinking of how games are conceived, developed, and played. Through the integration of generative tools deeply into the creative and technical workflows, developers can focus more on innovation and less on repetitive tasks, paving the way for a new era of game design where both creativity and scalability coexist harmoniously.

3. Foundations of video game development 3.0

LLMs have emerged as the central technology driving rapid content and code generation in game development. Through leveraging vast datasets and sophisticated architectures, models like GPT-4 have demonstrated the capability to autonomously generate dialogue options, quest lines, and even entire segments of code based on high-level natural language prompts [24]. These capabilities are not limited to textual outputs but extend to generating functional game levels and environments with intricate constraints. Research shows that LLMs can effectively generate game levels,

such as in the puzzle game Sokoban (2021) (**Figure 2**), where spatial and functional relationships are crucial [25]. The integration of LLMs into the game development pipeline enables developers to significantly reduce manual workload, allowing them to focus more on creative aspects while the AI handles the repetitive coding and content creation tasks.

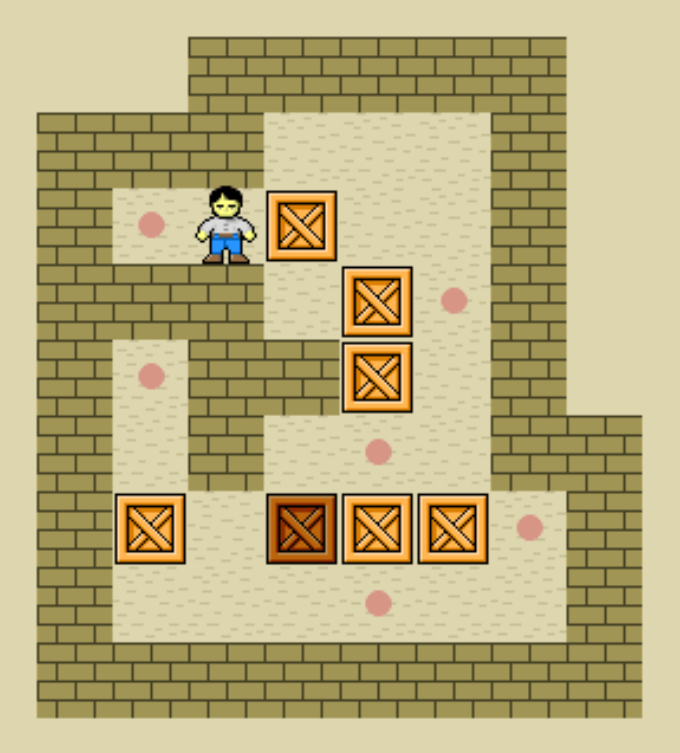

Figure 2. A sokoban puzzle and its solution (CC 3.0).

Figure 2 illustrates a Sokoban puzzle, highlighting the role of AI in procedural level design. In games like Sokoban, AI systems are employed to generate puzzle levels that adhere to specific spatial and functional constraints. The solution shown demonstrates the AI's ability to generate not only the puzzle itself but also ensure solvability within the game's mechanics. This figure is crucial in showing how AI-driven tools can create intelligent and adaptive game levels that enhance player experience through personalized challenges. These systems enable the dynamic generation of levels that scale in complexity based on player performance, showcasing AI's role in making gameplay both engaging and progressively challenging.

Moreover, the democratization of these tools through open-source platforms has had a transformative impact on both indie developers and large studios. Platforms like Godot's open-source engine have integrated generative capabilities that allow developers to experiment with cutting-edge techniques without being restricted by proprietary systems [26]. This democratization is further supported by contributions from the AI research community, which has made sophisticated tools more accessible. The open-source model has fueled innovation by enabling a broad range of developers to experiment with AI-driven content generation and procedural techniques. As a result, indie developers can now produce content of a quality comparable to larger studios, leveling the proverbial "playing field" in the industry.

Another way this technology is being used are in teacher models. Teacher models play a critical role in guiding AI systems to learn specific artistic styles or gameplay mechanics. Training AI on datasets with clearly defined parameters, such as the visual style of the game Cuphead (2007) or the narrative structure of The Witcher series (2007–2022), these models allow for the creation of assets that are consistent with a desired creative vision [27]. The refinement ensures that the AI-generated outputs are not only technically sound but also aligned with the thematic and stylistic requirements of a particular game. The use of teacher models represents a targeted approach to content generation, enabling developers to maintain creative control while benefiting from the efficiency of AI-driven processes.

4. Key components of video game development 3.0

4.1. Asset creation

The key components of video game development utilizing AI include asset creation, Game NPC AI, and Narrative Generation (**Table 1**). Generative Adversarial Networks (GANs) play a crucial role in automating asset creation by learning from existing datasets to generate game elements like character models, textures, and environments with minimal human input. GANs offer significant efficiency in terms of both time and cost, allowing developers to scale game assets across various artistic styles (e.g., pixel art, realism, or anime) consistently. By utilizing predefined parameters, these models maintain a uniform aesthetic across the game, ensuring a seamless visual experience. Developers can also fine-tune GANs for specific stylistic outputs, enhancing creativity while reducing manual labor. Additionally, the scalability of these tools enables the rapid expansion of game environments as needed without the tedious process of manual creation.

Table 1. Key components to video game development 3.0.

Component	Technology	Application
4.1 Asset Creation	GANs (Generative Adversarial Networks)	Generate game elements like character models, textures, and environments, automating manual labor and reducing costs.
4.1 Asset Creation	AI-driven rapid prototyping	Instant generation of game concepts and mechanics, allowing rapid exploration and iteration of ideas.
4.1 Asset Creation	AI-assisted level design	Generates and fine-tunes game levels by adjusting complexity based on player performance, improving engagement.
4.2 Game NPC AI	Reinforcement Learning (RL)	NPCs adapt their behavior dynamically based on player actions, creating personalized experiences.
4.2 Game NPC AI	Decision Trees	Allows NPCs to refine decision-making through real-time feedback, improving in-game interactions.
4.3 Narrative Generation	Large Language Models (LLMs)	Creates dynamic, context-aware narratives, adjusting dialogue and plot based on player input.
4.3 Narrative Generation	AI-driven processes	Real-time adaptation of game assets based on player choices, enhancing immersion and personalization.

AI-driven rapid prototyping significantly accelerates the creative process in game development by enabling the instant generation of game concepts. Leveraging advanced algorithms and generative models, developers can quickly explore novel

game mechanics and genres based on emerging trends and market needs. For instance, these systems have been used to rapidly prototype educational games, demonstrating their capacity to efficiently produce diverse and commercially viable digital solutions [28]. The ability to instantly generate and iterate on ideas allows developers to test concepts early and refine them before committing substantial resources, leading to more innovative and market-aligned game titles.

The integration of AI in the development process also allows for the rapid iteration of game mechanics, reducing development time and costs. The tools can automate playtesting and balance adjustments, enabling real-time modifications to combat systems or other game features based on player data and predefined parameters. This automated iteration not only improves game balance but also enhances player experience by ensuring that gameplay remains engaging and challenging throughout development [29]. By continuously refining mechanics through AI-driven feedback loops, developers can create more polished and enjoyable games with reduced manual input.

On the other hand, AI-assisted level design is another transformative application of AI in rapid prototyping. systems can generate and fine-tune game levels that are both challenging and enjoyable by dynamically adjusting level complexity based on player performance. Games like the 2D platformer Spelunky (2008) have benefited from AI-generated levels that adapt to player skill levels, offering a personalized and progressively challenging experience [30]. The use of the technology allows developers to focus more on creativity and innovation while ensuring that their levels remain well-balanced and enjoyable across different player demographics. But these tools can also assist with creating game art assets, significantly easing the burden on artists. Generative adversarial networks (GANs) and similar models can now autonomously produce high-quality character models, environmental textures, and other graphical elements across various artistic styles on demand. These tools allow for the rapid generation of diverse assets tailored to specific themes, enabling developers to experiment with different visual approaches without the need for extensive manual design [31]. For instance, GANs have been employed to produce procedurally generated game environments that are not only aesthetically consistent but also diverse enough to keep players engaged over extended gameplay sessions [32].

4.2. Game NPC AI

The AI-driven behavior of NPCs in modern games has evolved significantly. By integrating reinforcement learning (RL) techniques, NPCs can adapt to player actions, creating dynamic and evolving gameplay. For example, NPCs in Red Dead Redemption 2 can adjust their behavior in response to the player's moral decisions, leading to more personalized and immersive gameplay. This evolution in NPC AI is a significant improvement over traditional static behavior models, as AI-driven NPCs can generate real-time responses based on a set of learned behaviors. Additionally, decision tree algorithms in conjunction with RL enable NPCs to learn from players and refine their interactions as the game progresses, ensuring varied and more engaging gameplay over multiple sessions.

With level design and game assets being supported by these new tools, narrative

elements can now gain more attention from developers. In fact, LLMs have extended the capabilities of procedural content generation (PCG) by introducing context-aware narrative and dialogue elements into games. Unlike traditional PCG systems that focus primarily on structural aspects, LLMs like GPT-4 (**Figure 3**) add a layer of coherence and thematic consistency, enriching game narratives with meaningful dialogue and adaptable storylines. For example, in narrative-driven roguelikes, LLMs can generate dynamically evolving stories that respond to player actions, making each playthrough unique and tailored to the player's decisions [33]. To create responsive and personalized game experiences, systems are now capable of adapting game assets in real-time based on player interactions. This dynamic adjustment can affect visual themes, in-game music, and other elements, thereby enhancing immersion and engagement. For instance, games can modify their aesthetic or auditory components depending on player choices or gameplay style, allowing for a more customized and evolving experience [22]. These advancements in AI-driven content creation mark a significant shift towards more flexible and scalable game development processes, enabling developers to focus on crafting more engaging and player-centric experiences.

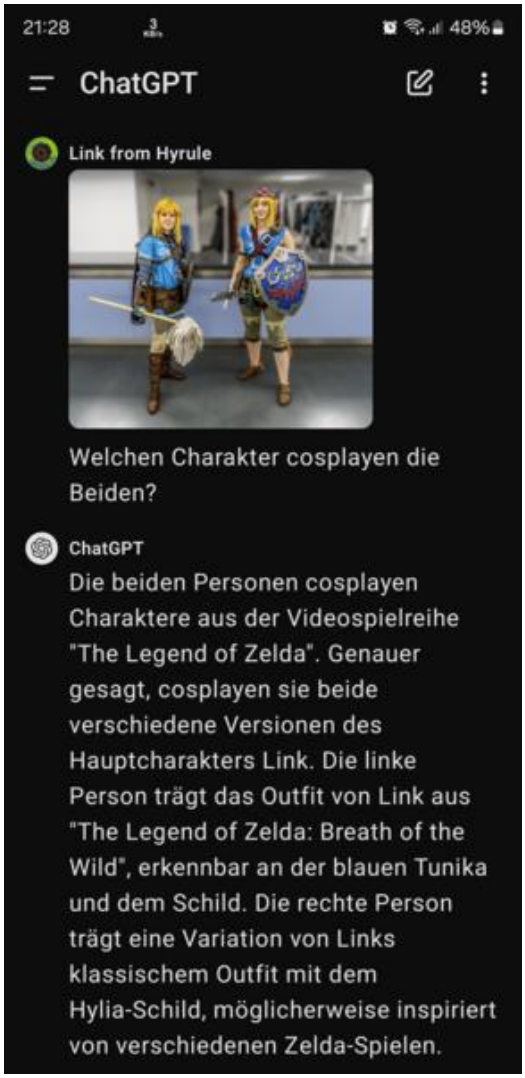

Figure 3. ChatGPT-4 cosplay example (public domain).

The next generation of video game development is heavily influenced by the integration of intelligent NPCs and dynamic game worlds. One of the most transformative aspects of this approach is the use of self-improving AI agents for NPCs. These agents evolve based on player interactions, allowing for more immersive and personalized gameplay experiences. Unlike traditional static NPCs, modern AI agents can adapt to player decisions, creating an evolving narrative that enhances engagement. For instance, in games like the action-adventure game Red Dead Redemption 2 (2018), NPCs adapt to the moral choices of the player, leading to consequences that feel more authentic and organic [34]. This shift from static behavior to adaptive systems reflects a broader trend toward more responsive and believable game environments.

4.3. Narrative generation

Dynamic narrative generation is another key component of these intelligent systems. Large Language Models (LLMs), such as GPT-4, are extensively used to generate dynamic, context-aware narratives in video games. However, controlling LLMs within typical game parameters requires reinforcement learning-based fine-tuning to ensure the generated content remains coherent and relevant to the game's objectives. LLMs can adjust narrative elements like dialogue and quest outcomes based on player input, enhancing the story's personalization. Through real-time feedback loops, LLMs continuously optimize the narrative to match player decisions, while avoiding over-generation or deviation from established plotlines. Leveraging AI-driven processes allows game narratives to now adapt in real-time to the choices made by players. In series like *The Elder Scrolls* (1994–2014), this enables quest lines and story arcs to evolve based on player actions, providing a unique experience with each playthrough [35]. This level of narrative flexibility not only enhances replay ability but also allows for deeper player immersion, as the story appears tailored to individual decisions and gameplay styles. Moreover, adaptive difficulty and personalized player experiences are becoming essential in maintaining a game's challenge while ensuring fairness. Survival-horror games like Resident Evil 4 (2005) exemplify this approach, where the AI adjusts enemy behavior and toughness in response to player performance, keeping the gameplay engaging without becoming frustratingly difficult [36]. These adaptive systems ensure that players of varying skill levels can enjoy the game, making the experience more accessible and tailored to individual preferences.

As noted, the integration of the technology as a creative partner in game design is revolutionizing the development process. The systems are now capable of generating insights based on data-driven analysis, suggesting alternate endings, new mechanics, and even optimizing level design according to genre trends and player preferences. This collaborative approach combines computational power with human creativity, leading to more innovative game designs [2]. The ability to offer real-time suggestions during brainstorming and prototyping enables faster iteration cycles, ultimately leading to a more refined final product. In addition to assisting with design, the tools are also transforming quality assurance (QA) and playtesting. Automated systems can now detect bugs, balance issues, and exploits much more efficiently than traditional methods. In Minecraft modding communities, for instance, AI tools help

identify and resolve issues in fan-made content, ensuring a smoother player experience while reducing the manual effort required from developers [37]. These enhancements streamline development processes and allow designers to focus more on creative tasks rather than troubleshooting technical issues.

In the era of community-driven development, as seen with modding communities, AI has become essential in efficiently integrating player feedback into the game development process. Modern systems can rapidly analyze vast amounts of data from player reviews, gameplay logs, and community discussions, identifying key areas for improvement and informing updates or patches. This approach is exemplified in first-person shooter (FPS) games like Fortnite (2017), where player feedback is parsed and analyzed to guide adjustments to in-game content, balancing, and feature additions [38]. By leveraging ML and natural language processing (NLP), these systems can distinguish between constructive feedback and noise, ensuring that only the most relevant insights inform development decisions. This continuous feedback loop helps developers maintain player satisfaction and engagement while keeping the game aligned with community preferences.

Another significant trend in community-driven development is the incorporation of user-generated content curated by AI. As more games embrace mods and other user-created elements, ensuring that these contributions fit seamlessly within the base game's framework is crucial. AI curation plays a pivotal role here, analyzing and selecting high-quality mods or user-generated content that align with the game's design philosophy while avoiding elements that could disrupt balance or coherence. In games like Skyrim (2011) and Minecraft (2011), systems help filter and integrate community-created content, maintaining the integrity and consistency of the original game while expanding its universe [39]. The approach not only empowers players to contribute creatively but also ensures that the overall quality of the gaming experience remains intact. Finally, these tools are increasingly being used to enhance and balance player-created content, offering amateur designers the means to produce high-quality levels, characters, and quests that are consistent with professional standards. 2D platformer games like Super Mario Maker (2015) demonstrate how AI can be employed to analyze and refine player-designed levels, ensuring they are not only fun but also balanced in terms of difficulty and gameplay flow [40]. Through automation of the more technical aspects of game design, these tools lower the barrier to entry for community members, encouraging broader participation in content creation while preserving the overall quality of the game.

5. Implementation methodologies

The AI-modular framework is designed to offer flexibility in game development by allowing developers to integrate components as needed, catering to various game genres. The approach enables the creation of specialized modules that can be tailored for specific elements such as combat, narrative, or character development. For instance, action RPG games like Mass Effect (2012) utilize modular components that manage complex decision trees for character interactions and story progression [41]. This level of modularity streamlines the development process, allowing teams to iterate quickly on various aspects of the game without needing to redesign entire systems. The

adaptability of these modular frameworks also ensures that they can be easily customized and extended as new game mechanics or features are introduced.

Decision trees are commonly used for in-game AI because they provide structured, hierarchical decision-making models that adapt dynamically to gameplay. Unlike static models, AI-driven decision trees allow NPCs to evaluate multiple conditions simultaneously, such as player proximity or environmental changes, and respond with appropriate actions, leading to more immersive and responsive gameplay. For instance, Auto-balancing uses AI to adjust the difficulty of a game in real-time. Games like FIFA use AI decision trees to evaluate player performance and dynamically adjust the difficulty of the AI-controlled opponents, ensuring balanced and engaging matches regardless of player skill level. This fine-tuning creates a seamless experience where the AI reacts to the player's proficiency, maintaining challenge without frustration. Similarly, in The Last of Us Part II, decision trees help AI opponents adjust their tactics based on the player's strategy, improving realism and engagement.

AI decision trees are also used to refine NPC interactions. For instance, in Horizon Zero Dawn, the AI relies on behavior trees to manage complex enemy behaviors. The AI system can make strategic decisions, such as when to attack or retreat, based on the player's actions. This fine-tuning process, driven by real-time data, makes NPCs more adaptive, creating a more challenging and unpredictable game environment. Behavior trees and finite state machines work together to evaluate multiple layers of decision criteria, making NPCs more realistic in their interactions. By using AI to handle these decision-making tasks, game developers can create adaptive and intelligent game mechanics that evolve based on the player's style, offering a more engaging and dynamic experience.

In addition, the integration of modular AI systems with existing game engines like Unity, Unreal, or Godot is crucial for widespread adoption. These engines are widely used across the industry, and ensuring compatibility allows developers to leverage existing workflows while incorporating advanced AI-driven content generation. For example, extensions such as EmbASP have been created to integrate declarative AI modules directly into the Unity game engine, enabling developers to use rule-based reasoning in real-time games [42]. By embedding AI capabilities into popular engines, developers can take advantage of plugins or APIs that simplify the inclusion of intelligent systems in game design, significantly enhancing the scope and quality of game content.

Another major advantage of modular AI frameworks is the ability to fine-tune models for specific game needs. APIs that facilitate this process make these tools more accessible to developers, even those without deep AI knowledge. These APIs provide interfaces that allow developers to tweak AI behavior, such as NPC decision-making or content generation, according to the unique requirements of their game. For example, in modular systems like SkyAI, developers can compose AI behaviors using predefined modules and customize them to fit different scenarios, reducing the time and expertise required to implement sophisticated AI features [43]. This modular and API-driven approach ensures that AI can be easily adapted and scaled, whether in indie projects or large-scale commercial games.

The integration of these real-time model updates based on player interactions is pivotal in maintaining dynamic and engaging gameplay experiences. AI systems can learn from player behavior and adjust game mechanics accordingly, leading to more responsive environments. For instance, in the online team-based games like Overwatch (2016), AI-driven matchmaking and balancing systems continuously adapt based on the skill levels and playstyles of participants, ensuring fair competition and sustained player satisfaction [44]. These real-time updates allow games to evolve organically, responding to player preferences while keeping the experience fresh and challenging. Correspondingly, A/B testing has become a standard practice in optimizing game content, especially when dealing with procedurally generated levels or new mechanics introduced by AI. By testing multiple versions of AI-generated content in parallel, developers can determine which version resonates better with players before rolling out updates or expansions. For example, in games like Destiny 2 (2017), different versions of AI-generated levels or quests are tested to gauge player engagement and satisfaction before a full release, minimizing the risk of negative reception [45]. This data-driven approach ensures that content is fine-tuned for optimal player experience.

However, as these tools become more integrated into game development, ethical considerations, particularly around bias and inclusivity, are increasingly important. AI-generated characters, narratives, and game mechanics must be scrutinized to ensure they do not reinforce harmful stereotypes or exclude certain demographics. Developers are adopting frameworks to mitigate bias in outputs, ensuring that games remain fair and inclusive for all players. Ethical AI design principles, such as those applied in character creation systems, prevent the propagation of biased representations, thereby fostering more diverse and equitable gaming experiences [46]. These considerations are crucial for maintaining the integrity of AI-driven systems in modern games.

But with increasing complexity of games, the days of storing the software on local hardware has been challenging. As such, cloud-based AI processing has become essential for managing the vast computational requirements of modern, complex game simulations. By offloading heavy computations to cloud servers, game developers can design more intricate game worlds without overburdening local hardware. For example, Microsoft Flight Simulator 2024 leverages cloud-based AI to simulate real-time weather patterns and global air traffic, enhancing the realism and immersion of the gameplay [47]. The architecture allows for detailed, large-scale environments that are continuously updated based on real-world data, creating dynamic and ever-changing game experiences.

While cloud computing enables complex simulations, edge computing is crucial for reducing latency in AI-driven interactions, ensuring smooth and responsive gameplay. Edge computing systems bring computational resources closer to the player, allowing for real-time processing that is especially important in fast-paced games like Call of Duty (2019), where milliseconds can make a difference in hit detection and AI-opponent behavior [48]. Through the distribution of tasks between cloud and edge servers, games can achieve low-latency responses while still benefiting from the scalability and power of cloud-based AI.

When considering intellectual property (IP), blockchain technology is increasingly being explored to manage the ownership and trading of AI-generated assets in games. In blockchain-based games like Axie Infinity (2018), players can own, trade, and even enhance in-game assets through decentralized networks, ensuring secure transactions and clear ownership rights. This integration of AI with blockchain enables a new level of personalization and economic engagement within games, allowing players to have tangible ownership over unique digital assets generated by AI algorithms [49]. The decentralized nature of blockchain ensures transparency and security, making it an attractive solution for the next generation of digital economies in gaming.

6. Case studies

The use of AI in procedural generation has revolutionized open-world game creation, allowing for the development of vast, complex environments with minimal human intervention. In action-adventure survival games like No Man's Sky (2016) procedural algorithms generate entire planets, each with unique ecosystems, landscapes, and flora, offering players nearly infinite exploration possibilities. AI-driven systems can handle tasks such as terrain generation, vegetation placement, and environment rendering, all while maintaining realistic physics and seamless transitions between various levels of detail [50]. These advancements have enabled games to provide players with dynamic and engaging worlds that evolve and diversify, thereby enhancing the overall gaming experience.

In fact, the potential of the technology in storytelling is exemplified in its ability to generate deeply personalized narratives that adapt in real-time to player decisions. In RPGs like Cyberpunk 2077 (2020) and The Witcher 3 (2015) systems are used to create branching narratives that evolve based on a player's choices, leading to multiple unique story outcomes. By analyzing player actions and preferences, these systems dynamically adjust the storyline, resulting in a personalized and immersive gaming experience [51]. The integration of AI into narrative design not only increases replay ability but also deepens player engagement by offering a storyline that feels uniquely crafted for each playthrough.

In competitive gaming, ensuring fairness and balance is paramount, and AI plays a crucial role in achieving these goals. Multiplayer online battle arena games like League of Legends (2009) use AI-driven analysis to monitor player performance and adjust game mechanics accordingly, such as fine-tuning champion balance or enhancing training bots. These systems continuously learn from player data, making adjustments to maintain a fair competitive environment while also improving training efficiency for professional players [52]. By leveraging AI, esports titles can provide balanced gameplay, fostering a competitive landscape that is both challenging and fair for all participants.

7. Challenges and ethical considerations

One of the core challenges in the implementation of AI-driven game development lies in maintaining a balance between technical assistance and human creativity. As these autonomous systems become more capable of generating content and code

autonomously, there is a risk that they could overshadow the unique creative contributions of human designers. Games like Journey (2012), which rely heavily on artistic expression and nuanced storytelling, exemplify the importance of preserving human creativity in the design process [53]. While the technology can be a powerful tool in augmenting creativity, there is a clear need to ensure that it remains a collaborative partner rather than the dominant force in creative decision-making [54]. This balance is crucial for sustaining the distinctive qualities that make each game unique and culturally significant.

As AI becomes more involved in content generation, there are also growing concerns about the potential for bias in AI-generated characters and narratives. Without proper safeguards, systems could inadvertently reinforce stereotypes or marginalize certain groups, leading to exclusionary content. Studios like Bioware have begun implementing frameworks to mitigate such biases, ensuring that AI-generated outputs reflect diverse and inclusive perspectives [55]. For example, character creation systems driven by AI must be carefully designed to offer a wide range of culturally diverse options while avoiding default settings that may perpetuate bias. The challenge lies not only in training models with diverse datasets but also in actively monitoring and adjusting outputs to prevent discriminatory or exclusionary outcomes.

The use of AI in analyzing player data raises important ethical questions about privacy and agency. In action-adventure games like *Watch Dogs* (2014), where player data significantly impacts the game world, there is a need for transparency and control over how this data is used [22]. Autonomous systems that adjust game content based on player behavior must ensure that players are fully informed about how their data is being utilized and have the option to control or opt out of certain data-driven features. Balancing the personalization benefits of the tools with respect for player privacy and autonomy is essential to building trust and maintaining ethical standards in AI-driven game development.

8. Future prospects

As gaming technology advances, the integration of AI with emerging platforms such as virtual reality (VR), augmented reality (AR), and brain-computer interfaces (BCIs) is expected to create increasingly immersive experiences. AI systems that dynamically adapt content in VR environments, as seen in FPS games like *Half-Life: Alyx* (2020), are paving the way for more engaging and responsive gameplay [56]. In AR, experiences like Pokemon GO (2016) leverage AI to seamlessly blend digital and real-world elements, providing a unique layer of interactivity. Looking ahead, the incorporation of BCIs into gaming could allow for direct interaction between player cognitive signals and in-game elements, enhancing immersion to unprecedented levels. These advancements highlight the role of this new technology in pushing the boundaries of player interaction within emerging technological landscapes.

The future of game design lies in creating persistent game worlds that evolve autonomously based on AI-driven interactions. In space-based, persistent world massively multiplayer online role-playing game (MMORPG) like EVE Online (2003), AI was already being used to manage complex ecosystems and economies, enabling a universe that operates independently of human input. As these tools continues to

develop, these worlds could become increasingly self-sustaining, with NPCs and environments that adapt in real-time to player actions, leading to game worlds that feel truly alive [57]. The ability to manage and evolve these spaces not only increases engagement but also allows for more complex and dynamic narratives that unfold over time, enhancing both the depth and longevity of gameplay.

These generative tools are set to democratize game development, making it accessible to a broader range of creators by lowering the technical barriers. Platforms like Roblox (**Figure 4**) and Dreams already empower users to design and share their own games, with AI playing a crucial role in simplifying content creation. AI-assisted design tools can help non-expert developers generate assets, scripts, and mechanics, allowing for high-quality outputs with minimal coding knowledge [58]. This democratization opens the door to a more diverse range of voices and ideas in the gaming industry, ultimately leading to richer and more varied content.

Figure 4. Figure from roblox doors (public domain).

9. Conclusion

The evolution towards video game development 3.0 signifies a transformative shift driven by the integration of advanced AI technologies throughout the development process. Central to this transformation are AI-driven rapid prototyping, dynamic asset creation, and intelligent NPC systems that respond to player interactions in real time. The capabilities offered by large LLMs and neural networks have redefined how games are designed, allowing for scalable solutions that cater to increasingly complex game environments. Through the automation of many of the repetitive and time-consuming tasks traditionally handled by developers, AI has unlocked new avenues for creativity, enabling faster iterations and more engaging content [59].

The broader implications of this paradigm shift extend beyond technical enhancements, reshaping the roles of developers, players, and even the business models underpinning the industry. The role of the technology in content generation not only enhances productivity but also democratizes game creation by providing non-experts with sophisticated tools to build high-quality games. The shift has the potential to redefine the relationship between creators and consumers, where players become co-creators and AI acts as a collaborative partner. The dynamic and adaptive nature of AI-managed game worlds offers a glimpse into the future, where game environments evolve independently, providing unique experiences each time they are played [60].

The continued evolution of AI in game development requires active engagement from all stakeholders, including developers, researchers, and players. Developers are encouraged to experiment with open-source tools and contribute to the growing ecosystem of AI-driven game development solutions. Researchers can explore the ethical and technical challenges associated with AI in gaming, while players can influence the direction of AI advancements by providing valuable feedback through their interactions with AI-driven systems. The future of gaming is poised for significant advancements, and collective efforts in exploring these possibilities will help shape a more inclusive, creative, and dynamic gaming industry.

Conflict of interest: The authors declare no conflict of interest.

References

1. Zackariasson P, Wilson TL. Paradigm shifts in the video game industry. Bengtsson M, ed. Competitiveness Review: An International Business Journal. 2010; 20(2): 139-151. doi: 10.1108/10595421011029857
2. Carbone JN, Crowder J, Carbone RA. Radically Simplifying Game Engines: AI Emotions & Game Self-Evolution. 2020 International Conference on Computational Science and Computational Intelligence (CSCI); 2020. pp. 464-472.
3. Charrieras D, Ivanova N. Emergence in video game production: Video game engines as technical individuals. Social Science Information. 2016; 55(3): 337-356. doi: 10.1177/0539018416642056
4. Puri R, Kung DS, Janssen G, et al. Codenet: A large-scale ai for code dataset for learning a diversity of coding tasks. Association for Computing Machinery; 2021.
5. Arnedo-Moreno J, Cooper KML, Lin D. Emerging Advanced Technologies for Game Engineering. ACM SIGSOFT Software Engineering Notes. 2024; 49(3): 37-41. doi: 10.1145/3672089.3672101
6. Nicolau M, Perez-Liebana D, O'Neill M, et al. Evolutionary Behavior Tree Approaches for Navigating Platform Games. IEEE Transactions on Computational Intelligence and AI in Games. 2017; 9(3): 227-238. doi: 10.1109/tciaig.2016.2543661
7. Partlan N, Soto L, Howe J, et al. EvolvingBehavior: Towards Co-Creative Evolution of Behavior Trees for Game NPCs. In proceedings of the 17th International Conference on the Foundations of Digital Games; 2022.
8. Gallotta R, Todd G, Zammit M, et al. Large Language Models and Games: A Survey and Roadmap. IEEE Transactions on Games; 2024.
9. Weber I. Large Language Models as Software Components: A Taxonomy for LLM-Integrated Applications. AIModels.fyi; 2024.
10. Shen Y, Shao J, Zhang X, et al. Large Language Models Empowered Autonomous Edge AI for Connected Intelligence. IEEE Communications Magazine. 2024; 62(10): 140-146. doi: 10.1109/mcom.001.2300550
11. Nijkamp E, Hayashi H, Xiong C, et al. CodeGen2: Lessons for Training LLMs on Programming and Natural Languages. CodeGen2; 2023.
12. Agarwal J, Shridevi S. Procedural Content Generation Using Reinforcement Learning for Disaster Evacuation Training in a Virtual 3D Environment. IEEE Access. 2023; 11: 98607-98617. doi: 10.1109/access.2023.3313725
13. Nasir MU, Earle S, Togelius J, et al. LLMatic: Neural Architecture Search Via Large Language Models And Quality

Diversity Optimization. In proceedings of the Genetic and Evolutionary Computation Conference; 2024.
14. Tan C, Zhang G, & Fu J. Massive editing for large language models via meta learning. Association for Computing Machinery; 2023.
15. Ali JM. AI-driven software engineering. Advances in Engineering Innovation. 2023; 3(1). doi: 10.54254/2977-3903/3/2023030
16. Spronck P, André E, Cook M, & Preuß M. Artificial and computational intelligence in games: AI-driven game design (Dagstuhl Seminar 17471). Dagstuhl Reports; 2018.
17. El Rhalibi A, Wong KW, Price M. Artificial Intelligence for Computer Games. International Journal of Computer Games Technology. 2009; 2009(1). doi: 10.1155/2009/251652
18. Lange D. Bringing Gaming; VR; and AR to Life with Deep Learning. In proceedings of the 25th ACM international conference on Multimedia; 23 October 2017.
19. Cutumisu M. Using behavior patterns to generate scripts for computer role-playing games [PhD thesis]. University of Alberta; 2009.
20. Carbonaro M, Cutumisu M, McNaughton M, et al. Interactive story writing in the classroom: Using computer games. In proceedings of DiGRA 2005 Conference: Changing Views: Worlds in Play 2005 Jan 1; 2005.
21. Pfau J, Liapis A, Yannakakis GN, et al. Dungeons & Replicants II: Automated Game Balancing Across Multiple Difficulty Dimensions via Deep Player Behavior Modeling. IEEE Transactions on Games. 2023; 15(2): 217-227. doi: 10.1109/tg.2022.3167728
22. Stephenson M. Generation and Analysis of Content for Physics-Based Video Games [PhD thesis]. The Australian National University; 2019.
23. Bakkes S, Spronck P, van den Herik J. Rapid adaptation of video game AI. 2008 IEEE Symposium On Computational Intelligence and Games; 2008.
24. Lin YC, Kumar A, Zhang WL, et al. Applications of Large Language Models in Data Processing: Innovative Approaches to Segmenting and Renewing Information. Available online: https://arxiv.org/abs/2311.16267 (accessed on 13 May 2024).
25. Todd G, Earle S, Nasir MU, et al. Level Generation Through Large Language Models. In proceedings of the 18th International Conference on the Foundations of Digital Games; 12 April 2023.
26. Nasir MU, Togelius J. Practical PCG Through Large Language Models. 2023 IEEE Conference on Games (CoG); 2023.
27. Akoury N, Yang Q, Iyyer M. A Framework for Exploring Player Perceptions of LLM-Generated Dialogue in Commercial Video Games. Findings of the Association for Computational Linguistics: EMNLP 2023; 2023.
28. Federley M, Sorsa T, Paavilainen J, et al. Rapid Prototyping of Mobile Learning Games. International Association for the Development of the Information Society; 2014.
29. King JF, Barton DE. Role of Simulation in Rapid Prototyping for Concept Development. Naval Engineers Journal. 1991; 103(3): 204-211. doi: 10.1111/j.1559-3584.1991.tb00950.x
30. Reyno EM, Carsí Cubel JÁ. Automatic prototyping in model-driven game development. Computers in Entertainment. 2009; 7(2): 1-9. doi: 10.1145/1541895.1541909
31. Yoo B, Kim KJ. Changing video game graphic styles using neural algorithms. 2016 IEEE Conference on Computational Intelligence and Games (CIG); 2016.
32. Bernardi L. Zermelo Game. International Journal of Serious Games. JMIR Publications; 2024.
33. Ashby T, Webb BK, Knapp G, et al. Personalized Quest and Dialogue Generation in Role-Playing Games: A Knowledge Graph- and Language Model-based Approach. In proceedings of the 2023 CHI Conference on Human Factors in Computing Systems; 19 April 2023.
34. Agis RA, Gottifredi S, García AJ. An event-driven behavior trees extension to facilitate non-player multi-agent coordination in video games. Expert Systems with Applications. 2020; 155: 113457. doi: 10.1016/j.eswa.2020.113457
35. Joyce L. Assessing Mass Effect 2 and Elder Scrolls V: Skyrim: Using collaborative criteria for player agency in interactive narratives. Journal of games criticism. 2016; 3(1).
36. Spronck P, Ponsen M, Sprinkhuizen-Kuyper I, et al. Adaptive game AI with dynamic scripting. Machine Learning. 2006; 63(3): 217-248. doi: 10.1007/s10994-006-6205-6
37. Aliaga C, Vidal C, Sepulveda GK, et al. Level Building Sidekick: An AI-Assisted Level Editor Package for Unity. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2023; 19(1): 392-399. doi: 10.1609/aiide.v19i1.27535

38. Rajapakshe U. Development Centric Player Feedback Analysis for Video Games: A Review. 2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS); 2019.
39. Thominet L. Tracing player experience. In proceedings of the 35th ACM International Conference on the Design of Communication; 11 August 2017.
40. Mozgovoy M. Analyzing User Behavior Data in a Mobile Tennis Game. 2018 IEEE Games, Entertainment, Media Conference (GEM); 2018.
41. Calimeri F, Germano S, Ianni G, et al. Integrating rule-based AI tools into mainstream game development. Springer International Publishing; 2018.
42. Williams B, Jackson J, Hall J, & Headleand CJ. Modular Games AI Benchmark. Springer International Publishing; 2018.
43. Dragert C, Kienzle J, Verbrugge C. Reusable components for artificial intelligence in computer games. 2012 Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques (GAS); 2012.
44. Nae V, Iosup A, Prodan R. Dynamic Resource Provisioning in Massively Multiplayer Online Games. IEEE Transactions on Parallel and Distributed Systems. 2011; 22(3): 380-395. doi: 10.1109/tpds.2010.82
45. Yang W, Zhang Q, Peng Y. A Dynamic Hierarchical Evaluating Network for Real-Time Strategy Games. Bevrani H, Shuhui W, eds. MATEC Web of Conferences. 2018; 208: 05003. doi: 10.1051/matecconf/201820805003
46. Richter V, Katzenbach C, Schäfer MS. Imaginaries of artificial intelligence. Handbook of Critical Studies of Artificial Intelligence; 2023.
47. Gao Y, Li Z. Deep Reinforcement Learning Based Rendering Service Placement for Cloud Gaming in Mobile Edge Computing Systems. 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC); 2023.
48. Tang X, Xu Y, Ouyang F, et al. A Cloud-Edge Collaborative Gaming Framework Using AI-Powered Foveated Rendering and Super Resolution. International Journal on Semantic Web and Information Systems. 2023; 19(1): 1-19. doi: 10.4018/ijswis.321751
49. Zhou H, Wang Z, Cheng N, et al. Stackelberg-Game-Based Computation Offloading Method in Cloud–Edge Computing Networks. IEEE Internet of Things Journal. 2022; 9(17): 16510-16520. doi: 10.1109/jiot.2022.3153089
50. Dunn IT. Procedural generation and rendering of large-scale open-world environments [Master's thesis]. California Polytechnic State University; 2016.
51. Robertson J, Young RM. Automated Gameplay Generation from Declarative World Representations. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2021; 11(1): 72-78. doi: 10.1609/aiide.v11i1.12790
52. Gemine Q, Safadi F, Fonteneau R, et al. Imitative learning for real-time strategy games. 2012 IEEE Conference on Computational Intelligence and Games (CIG); 2012.
53. Erden YJ. Could a Created Being Ever be Creative? Some Philosophical Remarks on Creativity and AI Development. Minds and Machines. 2010; 20(3): 349-362. doi: 10.1007/s11023-010-9202-2
54. Yang D. Designing mixed-initiative video games [Master's thesis]. Northeastern University; 2023.
55. Anantrasirichai N, Bull D. Artificial intelligence in the creative industries: a review. Artificial Intelligence Review. 2021; 55(1): 589-656. doi: 10.1007/s10462-021-10039-7
56. Wang J, Shi R, Xiao Z, et al. Effect of Render Resolution on Gameplay Experience, Performance, and Simulator Sickness in Virtual Reality Games. Proceedings of the ACM on Computer Graphics and Interactive Techniques. 2022; 5(1): 1-15. doi: 10.1145/3522610
57. Hong JH, Cho SB. Evolving Reactive NPCs for the Real-Time Simulation Game. Semantic scholar; 2005.
58. Ayas AY, Aydin H, Çetinkaya A, et al. Artificial Intelligence (AI) Based Self-Decision Making Character Development in Two-Dimensional Video Games (Turkish). Bilgi ve İletişim Teknolojileri Dergisi. 2023; 5(1): 1-19. doi: 10.53694/bited.1247338
59. Dwivedi YK, Hughes L, Ismagilova E, et al. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management. 2021; 57: 101994. doi: 10.1016/j.ijinfomgt.2019.08.002
60. Bilgram V, Laarmann F. Accelerating Innovation With Generative AI: AI-Augmented Digital Prototyping and Innovation Methods. IEEE Engineering Management Review. 2023; 51(2): 18-25. doi: 10.1109/emr.2023.3272799