The Impact of Artificial Intelligence on Game Development

Jan Philip Schultz Tsanas

December 5, 2023

1 Abstract

The integration of Artificial Intelligence (AI) into game development represents a significant advancement in the gaming industry, but it also poses numerous challenges and questions regarding its application and impact. This paper provides a literature review addressing the various roles of AI in enhancing various aspects of game development. The central focus of the study is to examine how AI technologies are being utilized to improve non-player character behavior, procedural content generation, and graphical improvements in games.

Key findings reveal that AI has enabled a significant leap in NPC behavior, allowing for more realistic, complex, and adaptive interactions, theoretically enhancing player immersion. In procedural content generation, AI could be used to create more dynamic and varied game environments, contributing to the uniqueness and replayability of games. Regarding graphical enhancements, AI-driven techniques are found to optimize rendering processes and improve visual fidelity, balancing high-quality graphics with performance efficiency.

The review also briefly mentions the challenges associated with the implementation of AI in game development, such as ethical considerations, technical limitations, and the balance between AI-driven automation and creative input. Overall, this paper highlights the transformative impact of AI in game development, offering insights into current applications, challenges, and future research potential.

2 Introduction

The evolution of AI and its integration into various sectors have started a new era of innovation and transformation. One of the most fascinating and rapidly evolving domains influenced by AI is the realm of game development. This working paper, intended as a literature review, delves into the various roles of AI in game development, focusing on three primary areas:

- How is AI used to enhance behaviour of non-player characters (NPCs) in games?
- How can AI be used to do procedural content generation like levels, quests and items?
- What is the role of AI in improving game graphic?

The first area of focus is using AI in augmenting the behavior of NPCs within games. NPCs play a crucial role in the gaming experience, offering interaction and challenges for a player's actions. Traditional approaches to NPC behavior have relied heavily on pre-scripted scenarios, limiting the depth and realism of interactions. With the help of advanced AI techniques, there is a shift towards creating NPCs that show more complex, adaptive, and realistic behaviors. This shift not only enhances player immersion but also significantly expands the capabilities of games. This paper reviews the latest research and developments in AI-driven NPC behavior, exploring how AI algorithms and models are being leveraged to create NPCs that can learn, adapt, and respond in increasingly human-like ways.

The second focal point of the paper is AI's role in procedural content generation, an area that has seen significant advancement with the application of AI technologies. Procedural content generation refers to the automated creation of game content like levels, quests, and items using manually created algorithms, reducing the need for manual input and allowing for greater scalability and diversity in game design.

However, manually created algorithms all have their weaknesses, as they are all limited to doing one particular task. With AI's involvement in this process, it opens up for richer content generation that understands the context in which it is being used and the content generation can adapt to new circumstances. This aspect of AI in game development opens up new possibilities for game design, allowing for personalized and ever-evolving gaming experiences that adapt to players' actions and preferences.

The third area examined in this paper is the impact of AI on game graphics. The relentless pursuit of realism and visual fidelity in games has always been a driving force in the gaming industry. AI has emerged as a key player in this quest, offering innovative solutions that enhance graphical quality while balancing performance. AI-driven techniques like real-time rendering optimizations, texture generation, and resolution enhancements are revolutionizing how games

are visually presented. These advancements not only contribute to more immersive and visually stunning gaming experiences but also help in optimizing performance, ensuring that high-quality graphics are accessible to a wider range of hardware specifications.

This literature review aims to provide a detailed overview of these three key areas, drawing insights from recent studies and developments in the field. It also discusses the challenges and potential future directions of AI in game development. By examining the impact of AI across different aspects of game development, this paper aims to contribute to a deeper understanding of the current state of the art and highlights areas for further research and exploration.

3 Methodology

In this section, I will describe the methodology used for this literature review. I will mention the databases and search terms used for finding the literature, the criteria for inclusion and exclusion of studies and the process which followed for screening and selecting the studies. This approach should ensure a replicable review of the literature on AI in game development.

3.1 Databases and search terms

In conducting this literature review, I utilized several reputable databases known for their extensive coverage of AI and machine learning research. These databases were ArXiv, ACM Digital Library and OpenAI.com. To ensure that I did not miss anything vital, I also included Google Scholar in my searches as a generic research database.

My search terms were carefully chosen to reflect the key aspects of the research questions. I used a combination of the following terms:

> *Language Model, Text to speech, Neural and Evolutionary computing, Denoising, Self supervised learning, Supersampling, Game, Real Time, Agents, Simulation, Human Behaviour, Procedural, Deep learning, Content Generation, Diffusion Model*

In addition to the previously mentioned databases, I also used Nvidia Research for finding relevant literature. This platform is a bit unique as it does not support traditional keyword searches. Instead, Nvidia Research organizes its papers into different research areas, such as *Artificial Intelligence and Machine Learning*, *Computer Graphics* and *Real-Time Rendering*, among others. This structure allowed me to explore each relevant research area independently and review the papers chronologically. Despite not being able to search, this method proved effective for finding relevant papers, particularly those related to AI's role in improving game graphics and generative AI.

Given that the "Nvidia Research" database is not searchable makes the literature search a bit less replicable, since search terms cannot be repeated by other

researchers. However in order to properly cover the state of the art in the fields of graphics and generative AI, I have decided to make this trade-off, as the alternative to not include Nvidia was a worse option.

The time frame for the literature search was restricted to studies published no earlier than 2020. Given the rapid pace of advancements in the field of artificial intelligence, particularly in game development, older papers can quickly become outdated. Limiting the search to more recent publications ensures that the review reflects the most current and relevant findings in the field.

3.2 Inclusion criteria

In order to select the appropriate articles for this literature review, certain guidelines were set up to filter through the extensive body of literature obtained through the search process. These guidelines were structured to guarantee that the selected articles are relevant to the research question and contribute useful knowledge to the study.

To begin, only articles written in English were taken into account. In addition, although scholarly articles often offer high-grade, trustworthy research, this study did not limit itself to scholarly literature. This choice was made to include potentially useful input from non-scholarly sources, such as project reports and technical summaries. This choice was made because the game development industry moves at a fast pace and often with a disregard for scientific research. This can make it difficult to get an accurate picture of the state of the art, if you restrict yourself to rigorously researched material.

The book *Game Development Research* [1] by Engström, provides an insightful commentary on the observed gap between academic research and practical game development. It states that *"good game development requires more than one disciplinary perspective; and, all existing research approaches that do not acknowledge this are likely to suffer from shortcomings"*.

Most importantly, it was not required for the articles to be focused exclusively on the gaming industry or game development. Instead, the main requirement was that the tools, methodologies, or findings demonstrated in the article could be used in game development in some form. This allowed for broader inclusion of pioneering AI techniques and strategies that might not traditionally be linked with game development but could still contribute to it.

For instance, an article discussing a unique machine learning method in the field of image identification might not explicitly reference game development. Nevertheless, if the method could potentially be used to enhance in-game visuals or the overall resolution of the game, such an article would be considered relevant to this review.

3.3 Screening process

When iterating though the found papers, a screening was done to ensure that the selected papers were relevant to the research question and sub-questions. The process was carried out in two stages: initial screening through title and abstract review, followed by a full-text review. For each of the sub-questions, a slightly different screening process was performed.

3.3.1 Papers relevant to enhancing NPCs

When looking for papers which could help enhance NPCs, I looked for studies that discussed AI methodologies, techniques, and tools used in training actors to behave intelligently in a given environment. This encompassed a wide range of AI applications, including reinforcement learning, meta-learning, and procedural content generation.

However, recognizing that the behaviour and perceived intelligence of NPCs are also heavily influenced by other aspects of game design, I also considered papers that discussed AI applications in enhancing the overall immersion of the game experience. For example, papers that discussed advancements in text-to-speech technologies were considered relevant, as improved dialogue and voice acting can significantly contribute to the believability and depth of NPC characters.

3.3.2 Papers relevant to procedural content generation

When screening literature for the "procedural content generation" topic, the term "procedural content generation" was interpreted broadly to capture any AI-driven techniques or tools that allow for creation of game content. "Content" was also broadly interpreted to not only describe levels or world, but to also consider anything from AI supported generation of dialog or textures to complete 3d models.

It's worth noting that while you could argue that studies and papers focused on text generation or dialogue generation as directly contributing to the "enhancing NPCs" sub-question, I have made the decision that such papers will be categorized under the "procedural content generation" sub-question. So, while they indirectly enhance the NPC's behaviour, their primary role is seen as generating content for the game, ensuring a dynamic gameplay.

3.3.3 Papers relevant to improving graphics

Games are becoming increasingly visually intricate, but they're also demanding a lot from our hardware. Higher resolutions, such as 4K and even 8K, are becoming the norm, and this places a substantial strain on rendering the game.

My primary focus during the screening process for papers related to graphics improvement was on performance enhancement and reducing rendering demands. With this in mind, I screened for papers that detailed AI techniques for denoising lower quality renders, making them clearer and more detailed, and also

screened for studies that explored supersampling of lower resolution renders. A requirement for all techniques were, that they should be possible in real-time to allow for use in games.

4 Findings

Following the title and abstract screening, I conducted a full-text review of the 15 papers that were selected across the three sub-questions for this study. This allowed me to examine in detail the methodologies and findings of the selected papers, ensuring that they provided substantial and applicable insights into the research topic. In the following sub-sections I will discuss the findings.

The themes in the papers are varied, covering a wide range of innovations in AI and gaming. To enhance NPC behavior, they include emergent tool-use by NPCs [2], generative emotional expression for more lifelike characters [3], real-time voice dialogue generation [4] and complex motor skill learning [5], large-scale skill learning for physically simulated agents [6] and adaptable agents for open-ended task spaces [7]

In the area of Procedural Content Generation, the papers explore adversarial training for robust dialogue systems [8], text-based 3D texture generation [9], creation of detailed 3D models from text prompts [10] and the use of Behavior Trees for dynamic level design [11].

Lastly, the work on Graphical Improvements ranges from neural network-based denoising techniques [12], efficient neural algorithms for supersampling [13], real-time denoising and supersampling combined [14], compression of 3D models [15], to specialized texture compression methods [16].

4.1 Enhanced NPCs

After the screening, 6 papers were selected [2, 3, 4, 5, 6, 7] for the sub-question regarding enhancing the behaviour of NPCs.

As the world of AI continues to progress, there is an increasing interest in creating tools, processes and techniques that allow for NPCs to be more than just scripted robots with predetermined responses and behaviours. The usage of AI advancements is redefining the capabilities of NPCs and how they influence in-game narratives.

The contribution of [2] lies in its exploration of multi-agent competition. In the study, researchers used a basic hide-and-seek setup as their environment and through countless cycles, the agents displayed not just growth but an suprising intuition to use tools and craft tactics that weren't set from the start. In most game setups, NPCs follow a set script, adhering to pre-set paths. Yet, this research suggests a horizon where NPCs, driven by fundamental goals, shape their own path and adapting to the ever-changing game world and reactions of the player.

Perhaps the most interesting part of this research is the emergence of tool use among the agents. This is not just about using a tool for its intended purpose but using tools in ways not intended to achieve the objectives. Translated to a gaming context, this could mean NPCs that can adaptively use in-game tools or elements, sometimes in ways that even game designers didn't foresee. Such adaptability can enhance gameplay depth, challenge players in unforeseen ways, and offer replayability as NPCs continuously adapt and surprise players in subsequent playthroughs.

Another way to improve NPC behaviour beyond problem-solving, is to enhance their emotions and reactions. [3] explores this with a combination of generative techniques and active reinforcement learning to spawn characters that mirror a broad range of human reactions. In the paper, a sandbox environment was designed, akin to 'The Sims', and populated it with twenty-five agents. These agents not only engage in daily activities like cooking and working but also form opinions, initiate conversations, and even plan events.

Using a large language model, the complete record of each agent's experiences are stored in natural language. This allows the agents to simulate their memories and using these memories they construct a higher-level and more abstract representation, which are stored in what the paper describes as "reflection trees". These reflections are then dynamically retrieved to guide the agent's behavior in various scenarios, making the behavior emergent rather than pre-scripted.

Now, imagine these nuanced behaviors complemented by voice. The research of [4] could be used to create NPCs that offer more than textual feedback, but without requiring static pre-made recordings, but rather voice dialogues generated on-the-fly. Think of sandbox games where players can have spontaneous chats with NPCs, with every exchange being context-driven and reflecting the NPC's evolving 'character.'

In the paper, they use a neural codec language model called VALL-E to generate high-quality speech from text. VALL-E was trained on a massive dataset of 60.000 hours of English speech between 700 speakers, a scale hundreds of times larger than other existing systems. This robust training enables the model to generate personalized speech based on a mere 3-second enrolled recording from a speaker.

The VALL-E model is a "Zero-Shot" model, meaning that it is able to handle tasks it hasn't been trained on. Which in this case, as a text-to-speech model, means being able to generate voices it hasn't heard during training, offering greater flexibility in applications like video games.

On a more physical skill front, [5] gives insights on possibilities of character controllers learning complex motor skills. In this case, juggling a soccer ball. What sets this work apart from related studies is the incorporation of multiple skills into one single policy. This provides a more centralized approach to teaching a model complex tasks, rather than focusing on individual skills in isolation. The model's ability to adapt and transition seamlessly between different juggling

skills, like foot juggling, chest juggling, and head juggling, indicates a level of complexity and adaptability that is currently unmatched.

A key component in the training of the model, is the use of an adaptive 'random walk'. An adaptive random walk is a process used to explore the state space efficiently. In the context of reinforcement learning, it serves to dynamically choose transitions during training which essentially, helps the model explore states and actions efficiently, leading to quicker and more effective learning.

Another paper which gives insights related to physical skills is [6]. This paper focuses on a framework that allows physically simulated agents to learn versatile and reusable skills. The approach leverages large-scale data to train a single policy network that can perform a wide range of tasks. This paper uses a data-driven paradigm for training the agents. The dataset used to train consists of motion clips that haven't been sorted or labeled for any specific task (e.g., walking, jumping, etc.). This lack of task-specific annotation allows the model to learn a wide range of behaviors without being limited to a specific subset of actions.

The framework used in the paper uses a discriminator which is part of the adversarial network. It attempts to differentiate between real motion data and the actions produced by the agent. This adversarial setup pushes the agent to generate actions that are indistinguishable from real, human-like motions, thereby enhancing its skill set. The efficiency of this method is also worth noting, as they mention that their framework can simulate 10 years of training in about 10 days, which highlights the scalability of their approach.

Lastly, the paper [7] explores how a reinforcement learning agent can adapt to a wide range of tasks at test time, even if these tasks were not part of its original training set. It introduces a new approach for training agents in an open-ended task space, utilizing a Transformer-based model.

Agents are placed in a procedurally generated 3D environment and given a simple goal. The paper then demonstrates that the agents, trained using the Transformer-XL model, can perform different tasks and adapt to different environments. The agents are able to strategize and react to both long-term and short-term actions from other agents or players. Agents are even able to cooperate to achieve a shared objective such as lifting an object together or organizing themselves into specific formations.

Unlike other models that may require extensive retraining to adapt to new tasks, the agents in this study are designed to adapt and learn new tasks within a timeframe comparable to human learning.

4.2 Procedural content generation

For the sub-question regarding how AI can be used to do procedural content generation, 4 papers were selected [8, 9, 10, 11] after the initial screening.

As the gaming industry evolves, games are becoming more and more immersive and complex. While procedural content generation is currently mostly used to randomize game elements, there is an interest in creating a deeper integration of advanced AI techniques. The application of machine learning and other AI methodologies can enable higher levels of detail, adaptability and realism in games using procedurally generated content.

In the quest for enhancing the capabilities of conversational AI models, [8] introduces an adversarial training framework, named Adversarial Turing Test (ATT), aimed at creating a robust model that can effectively discriminate between machine-generated and human-written responses in dialogue systems.

The paper introduces an iterative attack-defense mechanism between the generator and the discriminator, which allows both models to evolve over time. Unlike other adversarial training methods that focus on one-time training, this approach generates "unrestricted" adversarial examples through reinforcement learning. This provides a more robust training set for the discriminator.

The paper also incorporates a resetting strategy, meaning they reset the generator to its pre-trained state at the beginning of each attack phase. Unlike traditional Generative Adversarial Networks (GANs), this gives better adverserial examples and better performance. The model is also validated against a variety of unseen attackers, including other dialog models, to prove its robustness, a step not commonly taken in other adversarial training research.

Moving to a more visual aspect of content generation, [9] presents Text2Tex, a novel method for generating high-quality textures for 3D objects based on text prompts. The technique employs a pre-trained depth-aware image diffusion model to create textures from various viewpoints, while also addressing the issues of texture stretching and inconsistency commonly found in 3D texture synthesis. What sets this method apart is its "generate-then-refine" strategy. The model first generates an initial texture from a preset sequence of viewpoints and then refines it using an automatic viewpoint selection technique, which ensures complete texture coverage and high-quality textures across the object's surface

To check how good the created textures were, the authors used measurements like Frechet Inception Distance (FID) and Kernel Inception Distance (KID). These measurements showed that Text2Tex is really good at making high-quality textures. People in a study also liked the textures it made, confirming that it works well. And, it performed even better than other methods that use GANs and are trained for specific types of objects.

As games become more complex, there is a need for large quantities and diverse 3D meshes. The paper [10] aims to contribute with 3D generative models that synthesize textured meshes that can be directly consumed by 3D rendering engines. The paper introduces a tool (GET3D) that can create detailed and textured 3D models, like cars or animals, using only simple 2D pictures for its training. GET3D then converts these 3D models back into 2D images to be

used in training itself. This method significantly improves both the quality and the variety of the 3D models it can produce.

The tool can also be adjusted to generate the surface texture and lighting effects of the model and it can even be guided by text descriptors like "burned car" or "fluffy animal" to produce specific kinds of 3D models. GET3D demonstrates superior performance, both in terms of quality and diversity of the generated 3D shapes compared to existing models. It can handle high-resolution images up to 1024x1024. However, it does have limitations, such as the need for 2D silhouettes and known camera pose for training, which currently restricts its use to synthetic data.

Taking a step back to more traditional thoughts about "procedural generation", [11] introduces a novel approach to procedural content generation in video games by leveraging Behavior Trees (BTs). While BTs are traditionally used to model the behaviour of NPCs, this paper explores their use in modeling game design agents. The authors adapt BTs to generate game levels for titles like Super Mario, Mega Man and Metroid. So instead of NPC actions, the authors used design actions like 'Generate Segment' or 'Connect Rooms', thus creating modular, reactive and interpretable level generators.

It also introduces the concept of generic and blending BTs that can be applied across multiple games, and even suggests the use of BTs for dynamic level generation based on player behavior. This approach could provide a new pathway for creating adaptive and personalized gaming experiences, although this particular application remains largely theoretical in the paper.

4.3 Graphical improvements

For the sub-question regarding how AI can be used to improve games graphically, 5 papers were selected [12, 13, 14, 15, 16] after the initial screening.

As the demand for more realistic and visually compelling games increases, the balance between graphics and performance continues to be a challenge. This is another area where AI techniques can be useful. By refining image quality and increasing resolutions on-the-fly, these methods can allow for high graphical quality and smooth gameplay, without requiring an unrealistically high amount of resources.

One method for doing this involves producing low quality, noisy image renders and denoising them. The paper [12] presents such an approach. While traditional denoisers may be optimized for either surfaces or volumes, like fire or smoke, they generally do not perform well on scenes that contain both. The research in this paper deals with scenes that contain regular surfaces, but also volumentric elements. Using a specialized neural network-based denoising architecture, they separate the signal into volume and surface components, then filter each component individually.

The authors demonstrate that their approach outperforms existing state-of-the-art denoisers, such as Open Image Denoiser (OIDN) and hierarchical kernel prediction networks (HKPN), in complex scenarios, however it still has limitations. It has a dependency on high-quality motion vectors and is challenged by denoising high-density, low-sample count volumes. While the architecture is optimized for quality, it does have a computational and memory cost, making it less suitable for memory-intensive applications. Furthermore, the current implementation does not support depth of field or motion blur, which could limit its applicability in some games.

Supersampling is another approach to improving graphics while reducing the computational cost. The paper [13] introduces a neural algorithm for supersampling rendered content that is four times more efficient than existing methods while maintaining the same level of accuracy. The proposed algorithm shows a four-fold improvement in efficiency over the closest related work, without compromising on the level of accuracy. Moreover, the architecture scales well to larger magnification factors. The paper also benchmarks its approach against commercial solutions like DLSS 2.2, showing superior performance in terms of visual quality.

The method in the paper consists of two main parts: a "warping module" and a neural network. The warping module adjusts previous frames to align with the current one, accounting for things like camera movement. The neural network then takes this aligned frame and enhances its quality to produce a high-resolution output. The researchers created a new dataset called QRISP, specifically designed for improving video game graphics. This dataset includes various types of visual data like color, depth, and motion at different resolutions. They trained their neural network using this dataset, optimizing it to be both fast and accurate in enhancing lower-resolution images to higher quality.

Another paper that adresses issues of both denoising and supersampling in gaming graphics is [14]. The authors present a real-time neural network model designed for not only denoising but also supersampling. Denoising was previously considered too computationally expensive for real-time applications, but the authors claim that by combining it with neural supersampling, they bring their method to a more acceptable performance budget.

By combining two networks into one network, this method improves the quality of the rendered scenes but also makes the process more efficient. The model uses low-precision arithmetic, specifically 8-bit integers, to reduce computational and bandwidth requirements, making it feasible for real-time applications. One of the key limitations of this method is in the handling of sparsely sampled regions, particularly with respect to specular highlights. The paper mentions that the method struggles to reconstruct these accurately, which is a common issue across denoisers.

Another approach to improving performance in games is compressing data or reducing the level of detail of objects. The research of [15] introduces such a

method. In the paper, they compress 3D representations, such as scenes or objects, in a way that maintains high visual quality while reducing the amount of data needed for storage. This makes it easier to stream 3D content in real-time applications like games, as less data needs to be sent over the network. Additionally, the method allows for varying levels of detail, meaning that the same object can be rendered with more or less detail as needed, further optimizing performance.

The paper employs a method they call Vector-Quantized Auto-Decoder (VQ-AD), which is a decoder that replaces feature-vectors with simpler indices that point to a learned "codebook". The codebook is a collection of pre-defined feature-vectors that serve as a reference for the compressed data. So instead of storing the original vectors, they can store the indices instead. Using a neural network, the codebook and indices are trained together. The paper mentions memory and computational cost during training, as a significant limitation of this method. Another limitation is that the method introduces some visible high-frequency artifacts when compressing certain types of data.

Textures on objects are another part of games that are increasing in size and quality, requiring more and more storage and memory demands. The research of [16] propose a neural compression technique specifically designed for material textures. The authors introduce a method they call Random-Access Neural Texture Compression (NTC) which aims to provide high-quality texture representation while also being efficient in terms of storage and computational demands. What sets this work apart from existing industry standards like JPEG XL and AVIF is that it is designed to compress multiple texture channels and mipmap levels together, which allows the method to exploit correlations between channels and mip levels, leading to superior compression rates without sacrificing the ability to randomly access the texture data.

While this method performs exceptionally well for materials with multiple channels, its advantages are not so aparent for lower channel counts, such as single RGB textures. Additionally, the algorithm always decompresses all material channels, which may not be efficient in cases where only specific channels are needed. The method is also computationally more expensive than traditional hardware-accelerated texture filtering, although the authors argue that it remains practical for real-time applications.

5 Conclusion and discussion

Among the papers reviewed, a few studies stand out as particularly interesting. The study on emergent tool-use by NPCs [2] could change how we program and perceive NPC behavior, moving away from the traditional notion of NPCs as scripted entities, towards more dynamic and adaptive game worlds. The emergent tool-use and open-ended adaptability proposed by this and similar papers, could radically change gameplay, making it more unpredictable and engaging.

Regarding procedural content generation and graphical enhancements, the research of [10] and [15] stand out. The former introduces a tool that can generate 3D models guided by text-prompts, completely changing the way game assets could be created and the speed at which they are created. The latter offers a way to maintain high visual quality in 3D objects while dramatically reducing storage requirements, which is vital for real-time 3D rendering and streaming in games.

While the literature provides insights into theoretical possibilities, further research should focus on validating these techniques in real-world gaming scenarios, considering the perspectives and experiences of both developers and players.

These studies collectively point to a future where AI not only enhances the way games are played but also how they are prototyped and developed. However, more research is needed to understand not only the ethical implications but also the practical challenges of such advancements for game developers. As these unscripted behaviors gain prominence, the need arises to explore how developers can effectively control or guide them to align with the intended gaming experience.

There is also a potential gap in the research and evaluation from the perspective of the users and how they experience the procedurally generated content. How do players perceive and interact with an AI-driven world? As AI becomes more and more common in many different aspects of games, this could be the next area of further research and investigation.

This is also one of the points of Engström mentions in [1], as I mentioned in section 3.2.

References

[1] Henrik Engström. *Game Development Research*. University of Skövde, Sweden, ISBN 978-91-984918-8-3.

[2] Bowen Baker et al., 2020. *Emergent Tool Use From Multi-Agent Autocurricula*.

[3] Joon Sung Park et al., 2023. *Generative Agents: Interactive Simulacra of Human Behavior*.

[4] Chengyi Wang et al., 2023. *Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers*.

[5] Zhaoming Xie et al., 2022. *Learning Soccer Juggling Skills with Layer-wise Mixture-of-Experts*.

[6] Xue Bin Peng et al., 2022. *ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically*.

[7] Jakob Bauer et al., 2023. *Human-Timescale Adaptation in an Open-Ended Task Space*.

[8] Xiang Gao et al., 2023. *An Adversarially-Learned Turing Test for Dialog Generation Models*.

[9] Dave Zhenyu Chen et al., 2023. *Text2Tex: Text-driven Texture Synthesis via Diffusion Models*.

[10] Jun Gao et al., 2022. *GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images*.

[11] Anurag Sarkar et al., 2021. *Procedural Content Generation using Behavior Trees (PCGBT)*.

[12] Jon Hasselgren et al., 2023. *Joint Neural Denoising of Surfaces and Volumes*.

[13] Antoine Mercier et al., 2023. *Efficient neural supersampling on a novel gaming dataset*.

[14] Manu Mathew Thomas et al., 2022. *Temporally Stable Real-Time Joint Neural Denoising and Supersampling*.

[15] Towaki Takikawa et al., 2022. *Variable Bitrate Neural Fields*.

[16] Karthik Vaidyanathan et al., 2023. *Random-Access Neural Compression of Material Textures*.