

# Assessing Frontier LLMs in Solving Game Development Problems: Preliminary Findings Across Three Game Engines

Thiago Guedes Cruz de Vasconcelos<sup>1</sup>, Adams Amaral de Castro Filho<sup>1</sup>,  
Guadalupe Prado Saldanha Ribeiro<sup>1</sup>, Maria Andréia F. Rodrigues<sup>2</sup>,  
Nabor C. Mendonça<sup>2</sup>

<sup>1</sup> Centro de Ciências Tecnológicas  
Universidade de Fortaleza – Fortaleza, CE – Brazil

<sup>2</sup>Programa de Pós-Graduação em Informática Aplicada & GIRA Lab  
Universidade de Fortaleza – Fortaleza, CE – Brazil

{devasthiago, adamsconfilho, guadalupepradosr}@gmail.com,  
{mafr, nabor}@unifor.br

**Abstract.** *This paper evaluates three frontier LLMs (ChatGPT-4o, o3, and Gemini 2.5 Pro) against expert-rated human answers for 30 technical questions collected from online Q&A forums about three popular game engines: Unreal, Unity, and Godot. Our results reveal significant performance variance, with o3 demonstrating superior capabilities over Gemini 2.5 Pro and ChatGPT-4o. A primary weakness identified across all models was response completeness, where AI-generated answers often lacked the comprehensive detail of the human baseline. These findings suggest that although LLMs are powerful assistants, they are not yet a substitute for human expertise in engine-based game development tasks.*

## 1. Introduction

The recent advent of Large Language Models (LLMs) has fundamentally altered the landscape of software engineering, offering developers powerful new tools for code generation, debugging, and knowledge acquisition [Hou et al. 2024]. Game development, a uniquely complex discipline that merges real-time computation, intricate asset pipelines, and specialized hardware interaction, is no exception to this trend. Developers increasingly turn to models like ChatGPT and Gemini to navigate the complex APIs of modern game engines [Saei et al. 2025]. However, the reliability of LLM-generated answers in the highly specialized context of game development remains largely unexplored [Gallotta et al. 2024]. As reliance on these tools grows, a systematic evaluation of their performance against trusted human expertise becomes necessary [Xu et al. 2023].

To address this gap, this work systematically evaluates the responses generated by three frontier LLMs, OpenAI’s ChatGPT-4o and o3, and Google’s Gemini 2.5 Pro, against expert-rated human answers for 30 technical questions collected from online Q&A forums across three popular game engines: Unreal, Unity, and Godot. Our findings reveal significant performance disparities across models and engines, with o3 emerging as the clear superior model, followed by Gemini 2.5 Pro and ChatGPT-4o. In particular, completeness turned out to be a critical weakness for all models in several questions, with the AI-generated answers often lacking the comprehensive detail of the human-rated baseline. These results indicate that even though current LLMs can be powerful assistants to

game developers, their effectiveness is highly dependent on both the specific model and the technical domain (i.e., game engine) at hand.

The remainder of this paper is organized as follows: Section 2 positions our study within the context of relevant related work. Section 3 details our research methodology. Section 4 presents the results of our comparative study. Section 5 discusses the study’s practical implications and limitations. Finally, Section 6 offers concluding remarks and directions for future work.

## **2. Related Work**

We compare our work with two related lines of research: evaluating LLMs against human-expert knowledge, and using LLMs as technical assistants in game development.

### **2.1. LLMs vs. Human Expertise**

A significant body of work has focused on comparing the performance of LLMs like ChatGPT against the vast repository of human-generated knowledge on software development Q&A platforms such as Stack Overflow. These studies consistently find that although modern LLMs can produce factually correct and often more verbose answers, they may lack the nuance, conciseness, and contextual awareness present in top-rated human answers [Liu et al. 2023, Kabir et al. 2024]. Other research has focused on the impact of LLMs on the usage of these platforms, noting a decline in human engagement on Stack Overflow following the release of ChatGPT [del Rio-Chanona et al. 2023], which motivates the need to understand the quality of the AI-generated answers that are supplanting them. Our research contributes to this line by narrowing the focus to the highly specialized and complex domain of game development, which has not yet been thoroughly investigated [Gallotta et al. 2024].

### **2.2. LLMs in Game Development**

Saei *et al.* found that LLMs often struggle with game-specific tasks due to complex engine APIs and real-time constraints, leading to frequent hallucinations [Saei et al. 2025]. To mitigate this problem, models have been fine-tuned on domain-specific data. For example, Paduraru *et al.* fine-tuned Code Llama to generate unit tests for Unity in C++ and C# [Paduraru et al. 2024]. In procedural content generation, LLMs enable intuitive, prompt-based creation workflows [Maleki and Zhao 2024]. LLMs also enhance quality assurance by identifying bugs in gameplay footage [Zhang et al. 2025] and assessing player engagement [Melhart et al. 2025]. In asset management, major studios like Activision have employed LLM systems to retrieve assets from massive libraries using natural language queries, drastically improving discovery and workflow efficiency [Mikolas 2025]. Our work complements those efforts by systematically evaluating frontier LLMs on practical game engine problem-solving tasks.

## **3. Methodology**

The methodology employed in this research encompasses the selection of LLMs, game engines, and online forum questions, as well as the process of extracting and analyzing the responses produced by the selected LLMs for the selected questions.

### 3.1. Selected LLMs

We selected three frontier LLMs available in June-July 2025: OpenAI’s ChatGPT-4o and o3, and Google’s Gemini 2.5 Pro. These models were chosen due to their advanced knowledge and reasoning capabilities, and because they are commonly used to solve programming-related technical issues.

### 3.2. Selected Game Engines

For this study, we chose three popular game engines targeting distinct market segments [Video Game Insights 2025]: Unity, Unreal, and Godot. Unity, implemented in C#, is the long-standing market leader, particularly in the mobile and independent development sectors. Unreal, implemented in C++/ Blueprint, represents the standard for high-fidelity 3D graphics and AAA game production. Finally, Godot, also implemented in C++, is the leading open-source alternative.

### 3.3. Selected Online Forum Questions

We curated 30 questions (10 per engine) from the r/unrealengine, r/unity, and r/godot subreddits, posted between January and June 2025, ensuring that none of them was created before the selected LLMs’ knowledge cutoff dates. Reddit was chosen for its conversational format and use of visual media, which are essential for contextualizing game development issues. We filtered posts to find impactful problems with a clear, community-validated answer (i.e., the top-upvoted reply in the thread, often marked with “Solved” flair, with no conflicting higher-voted alternatives) to serve as our baseline. This small, curated set was chosen to keep the scope manageable for this preliminary study.

The main engine-related topics covered by each of the selected questions are described in Table 1. Although these questions are only a small sample of the type of questions game developers are posting in popular online forums, they are representative of broader issue patterns frequently observed in game development, such as save/load systems, performance optimization, scene structuring, and animation transitions.<sup>1</sup>

### 3.4. LLM Response Extraction and Comparative Analysis

We prompted each LLM with the selected questions using the *Question Prompt Template* shown in Figure 1. All prompts were submitted manually, via the models’ official web UI, using default settings for temperature and other parameters. When available, images were directly embedded in the prompts along with the question description; no image-to-text conversion was performed. Each question was submitted only once per model to reflect a single-interaction scenario.

We used the models themselves to assess all responses against the human baseline using the *Response Analysis Prompt Template* also shown in Figure 1. This template instructs the evaluating LLM to qualitatively analyze the given LLM responses for the given question and baseline answer according to three quality criteria: *Accuracy* (factual correctness), *Completeness* (coverage of essential aspects), and *Usefulness* (clarity and feasibility), with each rated *Satisfactory*, *Partially Satisfactory*, or *Unsatisfactory*. To mitigate bias, the LLM responses were anonymized in the analysis prompts (identified

---

<sup>1</sup>The questions’ transcripts, including the human-rated answers, are available in the paper’s companion repository at <https://anonymous.4open.science/r/llm-game-engines-8D2C/>

<p><b>Question Prompt Template</b></p> <p>A game developer is having an issue with the <i>[insert game engine name]</i> as reported below. Could you suggest possible ways for the user to solve that issue?</p> <p><i>[insert original question from online forum, including text and image(s)]</i></p>	<ul style="list-style-type: none"> <li>- Include specific examples or direct comparisons where necessary to justify your ratings.</li> <li>- After analyzing all chatbot responses for the question, provide a summary of the quality of the chatbots' responses for each evaluation criterium. Highlight which chatbot(s) had the best/worst response(s) overall (considering all criteria).</li> </ul>
<p><b>Response Analysis Prompt Template</b></p> <p><b>Objective:</b> Evaluate the quality of responses given by various AI chatbots to the technical question below, comparing them to a baseline answer.</p> <p><b>Evaluation Criteria:</b></p> <ol style="list-style-type: none"> <li>1. <i>Accuracy</i>: Assess the factual correctness of the chatbot's response. Identify any discrepancies compared to the baseline answer and note whether the chatbot's information aligns with trusted sources.</li> <li>2. <i>Completeness</i>: Check if the chatbot's response covers all essential aspects highlighted in the baseline answer. List any missing critical points.</li> <li>3. <i>Usefulness</i>: Assess the overall usefulness of the chatbot's response compared to the baseline. Consider whether the language is clear and well-structured, technical terms are appropriately explained, and the proposed solutions contain steps that are feasible and actionable.</li> </ol>	<p><b>Evaluation Template (for each response):</b></p> <ul style="list-style-type: none"> <li>- Accuracy: [Detailed analysis and rate]</li> <li>- Completeness: [Detailed analysis and rate]</li> <li>- Usefulness: [Detailed analysis and rate]</li> <li>- Overall Rating: [Overall rating considering all criteria]</li> </ul>
<p><b>Instructions:</b></p> <ul style="list-style-type: none"> <li>- Provide a detailed, comparative analysis of each chatbot response to the baseline answer based on the criteria listed above.</li> <li>- Qualitatively rate each chatbot response for each criterium using the following scale: <i>satisfactory</i> (the response meets or exceeds expectations), <i>partially satisfactory</i> (the response meets some expectations but has notable shortcomings), and <i>unsatisfactory</i> (the response fails to meet expectations).</li> </ul>	<p><b>Summary Template (for all responses):</b></p> <ul style="list-style-type: none"> <li>- Accuracy: [Summary of findings across all chatbots]</li> <li>- Completeness: [Summary of findings across all chatbots]</li> <li>- Usefulness: [Summary of findings across all chatbots]</li> <li>- Overall Best Response: [Which chatbot(s) performed best]</li> <li>- Overall Worst Response: [Which chatbot(s) performed worst]</li> </ul> <p><b>Question:</b> <i>[insert original question from online forum, including text and image(s)]</i></p> <p><b>Baseline Answer:</b> <i>[insert "best" (i.e., highest-rated) answer from online forum]</i></p> <p><b>Chatbot Response #1:</b> <i>[insert full response from Chatbot #1]</i></p> <p>...</p> <p><b>Chatbot Response #N:</b> <i>[insert full response from Chatbot #N]</i></p>

**Figure 1. Prompt templates used during the study.**

only as *Chatbot #1*, *Chatbot #2*, and so forth), and an LLM-assigned rating was only considered valid if at least two of the three evaluating LLMs agreed. We did not manually inspect responses for hallucinations or inaccuracies, focusing instead on LLM-based comparative rating.

## 4. Results

This section presents our comparative analysis of the three LLMs, based on consensus evaluations. Table 1 shows the per-question consensus ratings, while Table 2 summarizes overall performance by criterion and game engine.<sup>2</sup>

### 4.1. Qualitative Analysis

The detailed results in Table 1 reveal common weaknesses across all models. Some questions, like Unity's "Perspective / Mesh Distortion" (Q4) and "Version Upgrade Visual

<sup>2</sup>In the PDF version, one can access the full LLM responses and comparative analysis for each question by clicking on the question numbers in Table 1.

**Table 1. LLM Assessment Results**
















Game Engine	Q#	Topic	ChatGPT-4o			o3			Gemini 2.5 Pro		
			A	C	U	A	C	U	A	C	U
Unreal	01	NPC Save System Optimization	😊	😊	😊	😊	😊	😊	😊	😊	😊
	02	Optimizing Trace: Timer vs. Tick	😊	😊	😊	😊	😊	😊	😊	😊	😊
	03	Material Transition Blending	😊	😊	😊	😊	❗	😊	😊	😊	😊
	04	Actor Tick Scalability	😊	😊	😊	😊	😊	😊	😊	😊	😊
	05	Nanite & Lumen Explained	😊	😊	😊	😊	😊	😊	😊	😊	😊
	06	Multiplayer Replication Strategy	😊	😊	😊	😊	😊	😊	😊	😊	😊
	07	Thorough Save System Design	😊	😊	😊	😊	😊	😊	😊	😊	😊
	08	Saving Dynamic World State	😊	😊	😊	😊	😊	😊	😊	😊	😊
	09	Blending Player and AI Control	😊	😊	😊	😊	😊	😊	😞	😊	😞
	10	Version Control	😊	😊	😊	😊	😊	😊	😊	😊	😊
Unity	01	Fixing Mesh / Texture Stretching	😊	😊	😊	😊	😊	😊	😊	😊	😊
	02	Lighting Artifacts on Imports	😊	😊	😊	😊	😊	😊	😊	😊	😊
	03	Time Scale Reset In-Game	😊	😊	😊	😊	😊	😊	😊	😊	😊
	04	Perspective / Mesh Distortion	😞	😞	😊	😞	😞	❗	😞	😞	😊
	05	2D Sprite Clipping in 3D	😊	😊	😊	😊	😊	😊	😊	😊	😊
	06	Transparent Material Fix	😊	😊	😊	😊	😊	😊	😊	😊	😊
	07	Post-Processing Enhancements	😊	😊	😊	😊	😊	😊	😊	😊	😊
	08	Version Upgrade Visual Artifacts	😊	😊	😊	😞	❗	😞	😞	😊	😊
	09	Texture Seams Between Meshes	😊	😊	😊	😊	😊	😊	❗	😞	😊
	10	Editor Gizmo Icons in Scene View	😊	😊	😞	😊	😊	😊	😊	😊	😊
Godot	01	Uniform UV Mapping	😊	😊	😊	😊	😊	😊	😊	😊	😊
	02	Input Sequence Logic	😊	❗	😊	😊	😊	😊	😊	😊	😊
	03	Transparency Self-Intersection	😊	😞	😊	😊	😊	😊	😊	😊	😊
	04	Auto Collision from Mesh	😊	😊	😊	😊	😊	😊	😊	😊	😊
	05	Physics Overlap (Motion Modes)	😞	😞	❗	❗	❗	😊	😊	😊	😊
	06	Tunneling in Fast Collisions	😊	😊	😊	😊	😊	😊	😊	😊	😊
	07	3D UI Mouse Hover Detection	❗	😊	😊	😊	😊	😊	😊	😊	😊
	08	Lighting-Driven Visibility Control	😊	😊	😊	😊	😊	😊	😊	😊	😊
	09	Softbody Cloth Simulation	😊	😊	😊	😊	😊	😊	😊	😊	😊
	10	Diagonal Tile Pathfinding Gaps	😊	😊	😊	😊	😊	😊	😊	😊	😊
<b>Criteria:</b>			A = Accuracy   C = Completeness   U = Usefulness								
<b>Key:</b>			😊 Satisfactory	😊 Partially Satisfactory	😞 Unsatisfactory	❗ No Consensus					

Artifacts” (Q8), consistently led to Unsatisfactory ratings, showing that certain engine-specific issues remain difficult for all LLMs. Model-specific failures also emerged. Gemini 2.5 Pro underperformed on Unreal’s “Blending Player and AI Control” (Q9), while ChatGPT-4o frequently returned incomplete or unsatisfactory answers, especially on Unity questions like “Time Scale Reset In-Game” (Q3) and “Editor Gizmo Icons in Scene View” (Q10). ChatGPT-4o notably struggled with Godot questions, receiving mostly Partially Satisfactory ratings even for basic topics like “Uniform UV Mapping” (Q1) and “Diagonal Tile Pathfinding Gaps” (Q10), while the other two models performed well. The multiple instances of “No Consensus” accuracy cases, such as ChatGPT-4o and o3 on Godot “3D UI Mouse Hover Detection” (Q7) and “Physics Overlap (Motion Models)” (Q5), and Gemini 2.5 Pro on Unity “Texture Seams Between Meshes” (Q9), demonstrate the ambiguity in some LLM-generated answers, even from the perspective of other models.

## 4.2. Quantitative Analysis

**Overall Performance** As shown in Table 2, o3 was the best performer, with 86.7% of responses rated Satisfactory. Gemini 2.5 Pro followed with 48.9%, while ChatGPT-4o lagged at 35.6%. Most of their remaining answers were Partially Satisfactory, suggesting

**Table 2. LLM Consolidated Results (%)**

Game Engine	Evaluation	ChatGPT-4o				o3				Gemini 2.5 Pro			
		A	C	U	T	A	C	U	T	A	C	U	T
Unreal		90.0	10.0	60.0	<b>53.3</b>	100.0	90.0	100.0	<b>96.7</b>	50.0	10.0	30.0	<b>30.0</b>
		10.0	90.0	40.0	<b>46.7</b>	0.0	0.0	0.0	<b>0.0</b>	40.0	90.0	60.0	<b>63.3</b>
		0.0	0.0	0.0	<b>0.0</b>	0.0	0.0	0.0	<b>0.0</b>	10.0	0.0	10.0	<b>6.7</b>
Unity		50.0	30.0	50.0	<b>43.3</b>	80.0	70.0	80.0	<b>76.7</b>	60.0	30.0	60.0	<b>50.0</b>
		40.0	60.0	40.0	<b>46.7</b>	0.0	10.0	0.0	<b>3.3</b>	0.0	50.0	40.0	<b>30.0</b>
		10.0	10.0	10.0	<b>10.0</b>	20.0	10.0	10.0	<b>13.3</b>	20.0	20.0	10.0	<b>13.3</b>
Godot		10.0	10.0	10.0	<b>10.0</b>	90.0	80.0	90.0	<b>86.7</b>	70.0	60.0	70.0	<b>66.7</b>
		70.0	60.0	80.0	<b>70.0</b>	0.0	10.0	10.0	<b>6.7</b>	30.0	40.0	30.0	<b>33.3</b>
		10.0	20.0	0.0	<b>10.0</b>	0.0	0.0	0.0	<b>0.0</b>	0.0	0.0	0.0	<b>0.0</b>
All		50.0	16.7	40.0	<b>35.6</b>	90.0	80.0	90.0	<b>86.7</b>	60.0	33.3	53.3	<b>48.9</b>
		40.0	70.0	53.3	<b>54.4</b>	0.0	6.7	3.3	<b>3.3</b>	23.3	60.0	43.3	<b>42.2</b>
		6.7	10.0	3.3	<b>6.7</b>	6.7	3.3	3.3	<b>4.4</b>	10.0	6.7	6.7	<b>7.8</b>
<b>Criteria:</b>		A = Accuracy				C = Completeness				U = Usefulness			
<b>Key:</b>		 Satisfactory				 Partially Satisfactory				 Unsatisfactory			

frequent issues with response depth or completeness. Unsatisfactory ratings remained low overall (less than 8% for all models).

**Performance by Evaluation Criteria** Accuracy was generally high across models, with o3 leading at 90.0% Satisfactory rates, followed by Gemini 2.5 Pro (60.0%) and ChatGPT-4o (50.0%). Usefulness rankings largely mirrored this trend. Completeness was the main differentiator: o3 excelled (80.0% Satisfactory), while Gemini 2.5 Pro and ChatGPT-4o struggled with 33.3% and 16.7%, respectively.

**Performance by Game Engine** For Unreal, o3 achieved 96.7% Satisfactory across all criteria, while Gemini 2.5 Pro struggled (30.0%), and ChatGPT-4o landed in between (53.3%), with high accuracy (90.0%) but poor completeness (10.0%). Unity results showed more parity: o3 still led (76.7%), followed by Gemini 2.5 Pro (50.0%) and ChatGPT-4o (43.3%). However, this engine saw the highest incidence of Unsatisfactory ratings, with both o3 and Gemini 2.5 Pro achieving 13.3% on this rate. Finally, for Godot, o3 (86.7%) and Gemini 2.5 Pro (66.7%) performed well, but ChatGPT-4o showed a clear knowledge gap, with only 10.0% Satisfactory answers overall. This was the weakest performance observed across all models and engines.

## 5. Discussion

### 5.1. Practical Implications

First, developers should not treat LLMs as interchangeable; model choice should depend on the task and engine (e.g., o3 for Unreal; o3 or Gemini 2.5 Pro for Godot). Second, the high rate of Partially Satisfactory ratings, especially for completeness, presents a subtle risk of technical debt, meaning developers must treat LLM responses as a starting point, not a definitive solution. Finally, the presence of Unsatisfactory ratings across all models serves as a reminder that LLM-generated answers still requires rigorous validation against official documentation and established best practices.

## 5.2. Limitations and Threats to Validity

This study’s primary threat to *internal validity* lies in our use of LLMs for response evaluation, a practice often referred to as “LLM-as-a-Judge” [Zheng et al. 2023]. Despite mitigation strategies such as anonymization and majority consensus, this approach remains prone to bias and inconsistency. Bias may also arise from alignment and training data of the evaluating models, potentially favoring certain answer styles or content. The presence of “No Consensus” outcomes across all models directly reflects the ambiguity and limitations of using LLMs to assess one another. Additionally, each question was submitted only once per model, meaning we did not capture variability from repeated trials, and results may differ in subsequent runs due to the models’ inherent non-determinism. Further threats to internal validity stem from our reliance on specific prompt templates and evaluation instructions. Variations in wording, structure, or context could influence both the generated answers and their subsequent evaluations. Moreover, the human-rated baseline, while useful for anchoring evaluations, may not represent the optimal or only valid solution to each problem.

Our study also has limitations in *external validity*. The question set is relatively small (30 questions), and our evaluation focuses on a limited set of engines and frontier models. As such, the findings may not generalize to other types of problems, other LLMs, or future model versions, especially given the rapid pace of model development. In addition, the questions were sourced from online developer forums, which naturally bias the dataset toward problems considered difficult or ambiguous by practitioners. This may not represent the broader distribution of tasks game developers face in practice.

Finally, due to space constraints, we did not conduct a detailed analysis of the root causes behind the models’ Partially Satisfactory or Unsatisfactory responses, nor did we investigate the underlying reasons for the occasional rating divergences observed across models. We plan to address these limitations in future research.

## 6. Conclusion and Future Work

Our exploratory study demonstrates that LLM performance in engine-based problem solving in game development is highly variable across models and engines. In particular, o3 showed significant superiority over Gemini 2.5 Pro and ChaptGPT-4o, yet all models exhibited weaknesses in providing complete answers compared to human experts. These findings suggest that developers should treat these AI tools as fallible assistants, requiring a workflow of critical verification. Future work should prioritize validation by human experts, including a detailed failure mode analysis, and broaden the scope to more questions, game engines, LLMs, and prompting strategies.

### Artifact Availability

All artifacts produced as part of this research are publicly available at <https://anonymous.4open.science/r/llm-game-engines-8D2C/>.

### Acknowledgments

Maria Andréia F. Rodrigues and Nabor C. Mendonça are partially supported by CNPq’s grants 314776/2023-0 and 313558/2023-0, respectively.

## References

- del Rio-Chanona, M., Laurentsyeva, N., and Wachs, J. (2023). Are Large Language Models a Threat to Digital Public Goods? Evidence from Activity on Stack Overflow. *arXiv preprint arXiv:2307.07367*.
- Gallotta, R. et al. (2024). Large Language Models and Games: A Survey and Roadmap. *IEEE Transactions on Games*, pages 1–18.
- Hou, X. et al. (2024). Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79.
- Kabir, S. et al. (2024). Is Stack Overflow Obsolete? An Empirical Study of the Characteristics of ChatGPT Answers to Stack Overflow Questions. In *2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- Liu, J. et al. (2023). ChatGPT vs. Stack Overflow: An Exploratory Comparison of Programming Assistance Tools. In *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C)*, pages 364–373. IEEE.
- Maleki, M. F. and Zhao, R. (2024). Procedural Content Generation in Games: A Survey with Insights on Emerging LLM Integration. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 20, pages 167–178.
- Melhart, D., Barthet, M., and Yannakakis, G. N. (2025). Can Large Language Models Capture Video Game Engagement? *arXiv preprint arXiv:2502.04379*.
- Mikolas, L. (2025). Machine Learning Summit: Enhancing Development with LLMs and Multimodal Retrieval in ‘Call of Duty’. Game Developers Conference (GDC).
- Paduraru, C., Staicu, A., and Stefanescu, A. (2024). LLM-based methods for the creation of unit tests in game development. *Procedia Computer Science*, 246:2459–2468.
- Saei, A. D., Sharbaf, M., and Rahimi, S. K. (2025). Large Language Models for Game Development: A Survey on Automated Code Generation. In *First Large Language Models for Software Engineering Workshop (LLM4SE 2025)*. CEUR Workshop Proceedings.
- Video Game Insights (2025). The Big Game Engine Report of 2025. [https://vginsights.com/assets/reports/The\\_Big\\_Game\\_Engines\\_Report\\_of\\_2025.pdf](https://vginsights.com/assets/reports/The_Big_Game_Engines_Report_of_2025.pdf).
- Xu, B. et al. (2023). Are We Ready to Embrace Generative AI for Software Q&A? In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 1713–1717. IEEE.
- Zhang, B., Xu, M., and Pan, Z. (2025). Human-AI Collaborative Game Testing with Vision Language Models. *arXiv preprint arXiv:2501.11782*.
- Zheng, L. et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:46595–46623.