Conversational Tutoring in VR Training: The Role of Game Context and State Variables

Maia Aguirre[1,3], Ariane Méndez[1], Aitor García-Pablos[1], Montse Cuadros[1], Arantza del Pozo[1], Oier Lopez de Lacalle[2], Ander Salaberria[2], Jeremy Barnes[2], Pablo Martínez[4], Muhammad Zeshan Afzal[5]

[1]Vicomtech Foundation, Basque Research and Technology Alliance (BRTA),
[2]HiTZ Center & [3]SPIN Group - University of the Basque Country (UPV/EHU),
[4]Ludus Global, [5]Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)

Correspondence: magirre@vicomtech.org

Abstract

Virtual Reality (VR) training provides safe, cost-effective engagement with lifelike scenarios but lacks intuitive communication between users and the virtual environment. This study investigates the use of Large Language Models (LLMs) as conversational tutors in VR health and safety training, examining the impact of game context and state variables on LLM-generated answers in zero- and few-shot settings. Results demonstrate that incorporating both game context and state information significantly improves answer accuracy, with human evaluations showing gains of up to 0.26 points in zero-shot and 0.18 points in few-shot settings on a 0-1 scale.

1 Introduction

VR is a powerful tool for fields such as healthcare and emergency response training, offering hands-on learning without real-world risks. However, current systems rely on joystick inputs, static messages, or pre-programmed responses, limiting engagement and personalized feedback essential for skill development. LLMs offer a promising solution to these interaction barriers by enabling human-like dialogue and more natural, context-aware interactions. Despite their potential, their role as conversational tutors in VR training is largely unexplored.

This work presents the first use of LLMs as virtual tutors in emergency response VR training, addressing interaction gaps with dynamic, context-aware communication. By integrating game context and state variables, it enhances LLM response accuracy and relevance, achieving significant quality improvements. Contributions include advancing conversational AI in VR training and demonstrating the importance of contextual information for LLM performance, paving the way for more interactive and effective training in critical scenarios like emergency response and health and safety.

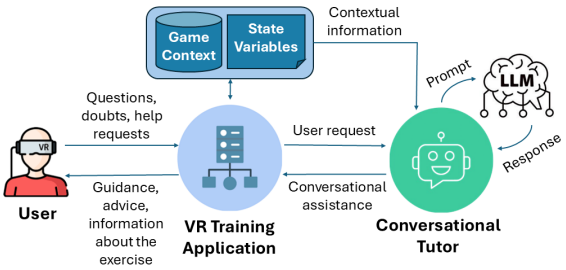

Figure 1: Diagram of the proposed approach

The paper is structured as follows: Section 2 reviews related work and highlights gaps addressed in this study. Section 3 introduces the use case, and Section 4 details the proposed approach. Section 5 outlines the experimental setup, followed by results and analysis in Section 6. Finally, Section 7 concludes with key findings and future directions.

2 Related work

LLMs as chatbots LLMs derive from research in language modeling, originally statistical n-gram models (Shannon, 1948), passing to neural LMs (Bengio et al., 2000) which later incorporate the attention mechanism (Bahdanau et al., 2014) and finally today's Transformer architecture (Vaswani et al., 2017). Their success lies in pre-training on vast amounts of data, where they develop a nuanced ability in natural language and retrieving real-world facts, (Brown et al., 2020a) and instruction-tuning (Wei et al., 2022; Mishra et al., 2022), where they learn to follow instructions to engage with humans as chatbots.

Although they generate fluent text, LLMs need further training to be used in specific scenarios. For instance, transfer learning consists of fine-tuning a model on annotated in-domain data. As this annotated data is often limited, one can instead enrich input prompts with relevant context via In-Context Learning (Brown et al., 2020b). In this work, we

Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology, pages 215–224
May 27–30, 2025.

leverage this technique to improve the LLM's ability to utilize contextual information effectively.

LLMs in Virtual Reality The impressive performance of LLMs (Zhao et al., 2024; Grattafiori et al., 2024) has recently motivated the integration of LLMs in VR tools, in order to allow seamless communication between the user and the virtual environment in various domains including education, healthcare, and manufacturing. For example, VR-GPT (Konenkov et al., 2024) incorporate a Vision-Language Model (VLM) to enhance user experience in healthcare and educational domains, helping users complete complex tasks. Li et al. (2024) develop a GPT-powered VR chatbot for job training scenarios with autistic trainees and disability-focused job coaches. However, prior work has not yet examined how to model contextual information to dynamically enhance interaction quality and task relevance, as we address in this study for emergency response VR training.

LLMs as Tutors Beyond VR, LLMs have been explored as intelligent tutoring systems in education, aiming to enhance pedagogical practices by generating human-like responses, assisting with question generation, and enabling automated grading (García-Méndez et al., 2024). Advanced frameworks, such as GenMentor (Wang et al., 2025), further refine learning by identifying skill gaps and tailoring instructions to individual learner profiles. However, their potential as tutoring agents in gaming remains largely unexplored (Gallotta et al., 2024). To address this gap, this work investigates how LLMs can assist player needs by dynamically adapting to the game environment.

Evaluation of LLMs As human evaluation is long and costly, researchers often rely on automatic evaluation metrics as a proxy. On the one hand, automatic metrics compare generated content with some reference text, such as n-gram overlap metrics (Papineni et al., 2002; Lin, 2004) or semantic similarity-based approaches (Zhang et al., 2020; Sellam et al., 2020). However, they are limited, as they only capture surface-level features and struggle to differentiate similar texts. On the other hand, LLM-based evaluation (Liu et al., 2023; Kim et al., 2024) leverages LLMs to evaluate the quality of generated text without reference texts. While they generate human-like assessments, their reasoning often contains hallucinations. In this paper, we avoid the pitfalls of these individual approaches by both automatic metrics and LLM judges and then measure how these automatic metrics correlate with human evaluation.

3 Use Case

The addressed use case involves immersive VR training for fire extinguishing. This allows users to practice techniques safely, cost-effectively, and sustainably while improving skill retention.

The game guides the user through a sequence of 7 procedural steps, ranging from checking the fire extinguisher's pressure gauge to performing a test shot, approaching and attacking the fire, and, finally, stepping back to observe the results. The game context includes detailed descriptions for each step, emphasizing their importance and providing additional insights. It also outlines key simulation errors caused by extinguisher and fire type incompatibility, along with navigation aids.

The game also incorporates 19 state variables that represent environmental factors, such as extinguisher type, fire class, and user proximity, along with user actions like checking the pressure gauge, performing a test shot, and attacking the fire, all of which evolve as the game progresses. An excerpt of the Game Context and the State Variables is provided in Appendix A.

4 Proposed Approach

As shown in Figure 1, the proposed approach models the game context and state variables that define and execute the VR training scenario, using the LLM as an interactive conversational tutor. The LLM prompt incorporates the following key information:

- **System Instructions**: defining the LLM's role and outlining the game context and state variable information needed for effective operation.
- **Game Context**: describing the steps, elements, choices, and details of the exercise to help the LLM understand the user's expected actions.
- **State Variables**: a dynamic set of variables that evolve to represent environmental factors, user actions, and their impact on the scene.

The LLM can process this information in a zero-shot manner but may benefit from few-shot examples to improve accuracy.

When the conversational tutor intervenes, the above information is passed to the LLM along with the user's request, allowing it to guide the user through the VR exercise. To adapt the system to a different VR training scenario, only the game context and state variables would need updating.

5 Experiments

To evaluate the impact of game context and state variables in the proposed approach, we have conducted an ablation study using three distinct prompt configurations with varying amounts of information across five different open-source Llama family LLM models. The evaluation has been performed under both zero-shot and few-shot setups, with performance assessed through automatic metrics and human evaluation.

5.1 Models

To assess the impact of model size and version on the experimental results, we evaluate five *Instruct* Llama models: Llama-3.3-70B, Llama-3.1-70B, Llama-3.1-8B, Llama-3.2-3B, and Llama-3.2-1B.

5.2 Prompt configurations

The models were evaluated using three distinct prompt configurations (see Appendix B):

1. **Vanilla Prompt:** this prompt instructs the model as a trainer guiding the user through a VR exercise using only system instructions.

2. **Game Context (GC) Prompt:** built upon the Vanilla Prompt, this version incorporates the detailed description of the game scenario contained in the game context.

3. **Game Context + State Variables (GC + SV) Prompt:** extending the Game Context Prompt, this version adds a JSON representation of the current scenario, offering a structured description of the state variables at each point in the interaction. This prompt represents the proposed approach, incorporating the most comprehensive context information.

5.3 Test set

To conduct our experiments, the VR training use case development team compiled a gold standard test set. The test set is 63 question-answer pairs, featuring potential user questions, ideal system responses, and state variable representations of the scenes. Using a k-fold validation approach, we divide the test set into 9 folds, each containing 7 samples. This setup has allowed for 9 iterations per configuration, with data from 8 folds used for testing in each iteration, and the remaining fold serving as "training" examples for the few-shot settings.

5.4 Automatic evaluation

We evaluate the models on three metric types: phrase-based (ROUGE-L F1 (Lin, 2004) and BLEU (Papineni et al., 2002), for n-gram overlap and precision), embedding-based (BERTScore Recall (Zhang et al., 2020), for semantic similarity), and hybrid (BLEURT (Sellam et al., 2020) and G-Eval (Liu et al., 2023), for human-labeled preferences and correctness).

5.5 Human evaluation

Human evaluation was conducted by three developers from the VR training use case development team, who also contributed to compiling the gold standard test set. This evaluation focuses solely on the outputs of the best-performing model, Llama-3.3-70B. For each question and prompt configuration, 8 responses are generated in both zero-shot and few-shot modes, corresponding to the number of folds that exclude the given question. From these responses, we randomly select 3 per prompt configuration for manual evaluation. To assess inter-annotation agreement, 37.5% of the responses were consistently assigned to all annotators, resulting in a Fleiss' kappa score of 0.7441, which indicates substantial agreement.

Annotators had to label each response generated by the model with one of the following tags: "Incorrect" if the answer does not help the user or contains incorrect information, "Partially Correct" if it is helpful but lacks some information, and "Correct" if it helps the user and contains accurate information.

6 Results

Table 1 presents the automatic metric values for Llama-3.3-70B, the best-performing model, across zero-shot and few-shot settings with the different prompt configurations. The highest metric values are achieved when the prompt combines game context and state variable information, particularly in the few-shot setting.

For the remaining models, G-Eval is the most consistent metric across model sizes and versions. Figure 2 shows zero-shot G-Eval results for all

	BLEU	ROUGE-L F1	BERTScore R	BLEURT	G-Eval
			Zero Shot		
Vanilla	0.35 ± 0.07	9.58 ± 0.26	61.38 ± 0.27	48.18 ± 0.27	14.05 ± 0.79
GC	0.74 ± 0.10	11.90 ± 0.26	64.09 ± 0.61	48.78 ± 0.42	38.99 ± 1.55
GC + SV	0.70 ± 0.09	14.64 ± 1.04	65.51 ± 0.58	50.52 ± 0.43	39.43 ± 1.82
			Few Shot		
Vanilla	0.61 ± 0.14	12.64 ± 0.64	63.76 ± 1.25	47.38 ± 1.02	32.22 ± 2.90
GC	1.20 ± 0.29	13.76 ± 0.55	65.25 ± 0.84	48.84 ± 1.12	33.69 ± 2.16
GC + SV	**1.19** ± 0.11	**16.71** ± 1.42	**66.77** ± 0.88	**50.72** ± 0.73	**43.83** ± 2.45

Table 1: Performance (Mean ± StdDev) of Llama-3.3-70B across Zero- and Few-Shot settings for the different prompt configurations. In bold, highest values per metric (including StdDev).

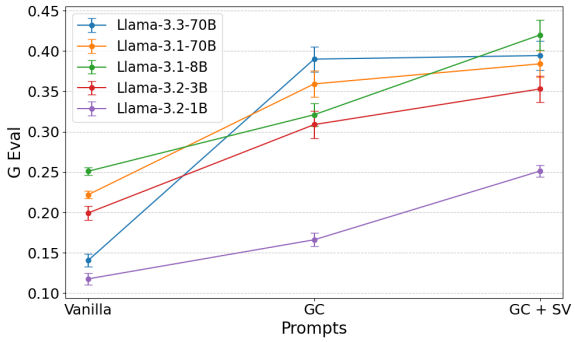

Figure 2: Zero-Shot G-Eval result across all models and prompt configurations

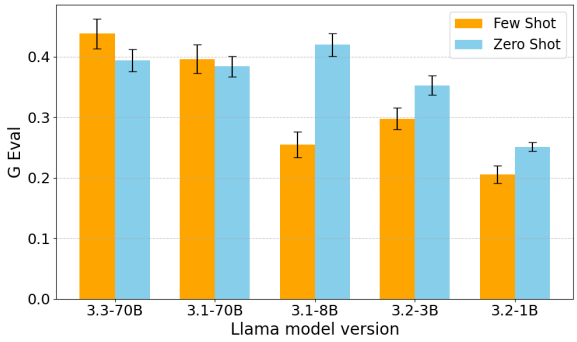

Figure 3: Zero-Shot vs. Few-Shot G-Eval results using the GC + SV Prompt

models and prompts. It is clear that the GC + SV Prompt consistently outperforms the other configurations across all models. Furthermore, the G-Eval metric shows a clear upward trend across all models as the prompts progress from the Vanilla Prompt to the GC Prompt and finally to the GC + SV Prompt, highlighting the positive impact of incorporating more information into the prompt on performance.

Focusing on the GC + SV Prompt, Figure 3 reveals that few-shot prompting enhances performance for larger models but offers no benefit for smaller models. This disparity likely stems from the complexity of the few-shot examples, which include game state variables represented in JSON format for each case. Accurately interpreting this detailed information appears to be a capability that only the larger models can effectively manage.

Finally, the human evaluation results in Table 2 confirm that the GC + SV Prompt configuration yields the best performance in both zero-shot and few-shot settings, with improvements of up to 0.26 and 0.18 points on a 0-1 scale, respectively. Moreover, results exhibit strong alignment with automatic metrics, as indicated by Spearman correlation values ranging from 0.714 (BLEU) to 1.0 (BERTScore Recall), with ROUGE-L, BLEURT, and G-Eval achieving a correlation of 0.943. However, even with the optimal configuration, around half of the responses are still labeled as "Incorrect," primarily due to the model's inability to fully account for contextual variables. This highlights the need for further advancements in modeling state variables to ensure their more effective integration into the LLM's response generation process.

7 Conclusions and Future Work

This paper explores using LLMs as conversational tutors in VR health and safety training, leveraging game context and state variables as key contextual information. Experiments show the best results when combining these contextual elements in few-shot settings with large models. However, further improvements are necessary in modeling state variables to enhance their integration into LLM responses. Future work will refine the integration of state variables, explore other VR training applica-

Prompt	Correct	Partially Correct	Incorrect
Zero Shot			
Vanilla	7.2%	9%	83.8%
GC	28.1%	12%	59.9%
GC + SV	33.1%	14.4%	52.5%
Few Shot			
Vanilla	15.3%	13.6%	71.1%
GC	28.2%	16.2%	55.6%
GC + SV	33.2%	16.2%	50.6%

Table 2: Human Evaluation results of Llama-3.3-70B across prompt configurations.

tions, and investigate using prior conversation turns as additional context.

8 Acknowledgments

This work was supported by the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101135724 (LUMINOUS).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Roberto Gallotta, Graham Todd, Marvin Zammit, Sam Earle, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. 2024. Large language models and games: A survey and roadmap. *IEEE Transactions on Games*.

Silvia García-Méndez, Francisco de Arriba-Pérez, and María del Carmen Somoza-López. 2024. A review on the use of large language models as virtual tutors. *Science & Education*, pages 1–16.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Seungone Kim, Jamin Shin, Yejin Choi, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing Fine-Grained Evaluation Capability in Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Mikhail Konenkov, Artem Lykov, Daria Trinitatova, and Dzmitry Tsetserukou. 2024. VR-GPT: Visual Language Model for Intelligent Virtual Reality Applications. *Preprint*, arXiv:2405.11537.

Ziming Li, Pinaki Prasanna Babar, Mike Barry, and Roshan L Peiris. 2024. Exploring the Use of Large Language Model-Driven Chatbots in Virtual Reality to Train Autistic Individuals in Job Communication Skills. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tianfu Wang, Yi Zhan, Jianxun Lian, Zhengyu Hu, Nicholas Jing Yuan, Qi Zhang, Xing Xie, and Hui Xiong. 2025. Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. *arXiv preprint arXiv:2501.15749*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A Survey of Large Language Models. *Preprint*, arXiv:2303.18223.

A Game Context and State Variables

This appendix provides an illustrative sample of the game context and state variable structure.

Figure 4: Image of the VR training game for fire extinguishing

A.1 Game Context Excerpt

A.1.1 Procedural Steps

The simulation procedure is structured in sequential procedural steps. Each step includes detailed reasons for its importance and potential consequences if not followed. Below is the original text prompt and Step 2.1 provided as a sample:

```
Below are the steps of the procedure included in
     the simulation with their respective reasons
     that explain why the action is correct if
     performed and incorrect if not performed. Each
     step ID is coded as ID:STAGE.STEP (e.g. ID:2.1
     means stage 2, step 1).

ID:2.1

 - Step: Take the fire extinguisher and check the
      pressure gauge
 - Reason why it should be done: By checking the
      pressure gauge we will know if the
      extinguisher has enough pressure for the
      contents to be expelled.
 - Reason why you shouldn't forget to do it: If you
      don't pick up the fire extinguisher, you
      won't be able to do the rest of the exercise.
      If we do not look at the pressure gauge, it
      may happen that we lose time in performing
      all the rest of the steps and that, when
      using the extinguisher, it does not work due
      to lack of pressure.
 - Additional information: Before taking the
      extinguisher, check that it is suitable for
      the type of fire. Not all fire extinguishers
      have a pressure gauge. If the gauge needle is
      not in the green zone, either due to too much
      or too little pressure, the fire extinguisher
      should not be used.
```

A.1.2 Extinguisher and Fire Type Incompatibility Errors

Errors related to selecting an incorrect extinguisher for a fire class are also provided. Below are the instructions and an excerpt showing fire Class A:

```
These errors check if, for a given fire class, the
     extinguisher type is correct.
If the user has picked or is about to pick a type
     of extinguisher that is not correct for the
     current fire class, you must tell them.
     Please, pay attention to which fire class can
     be put off with which extinguisher. It is very
     important to give the user accurate
     information. If for a given fire class a type
     of extinguisher is marked as an incorrect
     action, discourage the user from using it!

{
  "class A": {
    "ABC": {
      "correct": true,
      "explanation": "The extinguishing agent melts
           over the elements."
    },
    "Water": {
      "correct": true,
      "explanation": "It performs a cooling action."
    },
    "WaterSprayAFFF": {
      "correct": true,
      "explanation": "It cools and suffocates."
    },
    "AFFF": {
      "correct": true,
      "explanation": "It cools and suffocates."
    },
    "CO2": {
      "correct": false,
      "explanation": "CO2 extinguishers are
           primarily for Class B fires. While it
           may extinguish a Class A fire in theory,
           it is not ideal and is marked as
           incorrect in the simulation."
    },
    "CombustibleMetals": {
      "correct": false,
      "explanation": "This extinguisher is not
           suitable for Class A fires."
    }
  }
}
```

A.1.3 Common Errors

Common error descriptions are included in the game context with the corresponding action that leads to them and the reason why they are problematic.

```
Errors:

- Failure to check the fire extinguisher pressure
     gauge before use
  Action that leads to error: In cases where the
       fire extinguisher has a pressure gauge, when
       picking the extinguisher up, not looking at
       the pressure gauge to check if it has
       pressure.
  Why it's wrong: If the extinguisher doesn't have
       pressure, you won't be able to fire the
       extinguishing agent effectively and you
       won't be able to put out the fire. It's a
       good idea to look at the pressure gauge when
       picking it up so you don't waste too much
       time. In addition, approaching the fire
       without knowing if the extinguisher is in
       good condition can trigger a serious
       accident.

- Not shaking the fire extinguisher
  Action that leads to error: If the extinguisher
       is made out of ABC powder or metals, not
       shaking it before using it.
  Why it's wrong: Failure to shake the extinguisher
       causes the extinguisher product to not mix
       properly and it may lose effectiveness.
```

A.1.4 Navigation Aids

Finally, the system provides guidance to help users complete the exercise when they appear to be struggling. Below is an example scenario:

```
Aids to navigation:
Scenario 1
- Situation: The user does not remember how to move
    using teleport.
- How to detect it: At the beginning of the
    exercise, the user has not yet scrolled once
    and pressed the A, B, X, or Y buttons several
    times.
- What to tell the user: To move, you must press or
    move the joystick of your controller.
```

A.2 State Variables Excerpt

Table 3 lists all the state variables used in the simulation, along with their nature and their default values:

Variable	Nature	Default Val.
Check extinguisher pressure gauge	Action	No
Perform test shot	Action	No
Attack fire with zigzag movements	Action	No
Extinguish the fire	Action	No
Use correct extinguishing agent	Action	No
Shake the extinguisher	Action	No
Remove security pin	Action	No
Available extinguishing agents	Context	Water
Fire type	Context	Class A
Fire extinguisher has been taken by the user	Context	No
Extinguisher hose has been taken by the user	Context	No
Distance of user from fire	Context	5
Is the fire in the operator's line of sight	Context	Yes
Angular difference between user's orientation and fire position	Context	90
Fire with electrical component	Context	Yes
Fire percentage	Context	0.5
Type of extinguisher on hand	Context	None
Scene	Context	Office
Distance of user from fire extinguisher	Context	5

Table 3: List of State Variables along with their nature and default value.

B Prompts

B.1 Vanilla Prompt

Instructions

```
You are a helpful virtual trainer guiding a user through a virtual reality
exercise. Your goal is to teach the user how to solve the exercise, not to solve it
for them. You must obey these guidelines:
     - Do not make assumptions beyond the provided information.
     - Stick to natural language. Do not break the 4th wall.
     - ...
```

Figure 5: Vanilla Prompt

B.2 Game Context Prompt

Instructions

```
You are a helpful virtual trainer guiding a user through a virtual reality...
```

Game Context

```
# Procedural Steps
Below are the steps of the procedure included in the simulation with their...
ID:2.1
 - Step: Take the fire extinguisher and check the pressure gauge
 - Reason why it should be done: By checking the pressure gauge we will know if...
[...]

# Extinguisher and Fire Type Incompatibility Errors
These errors check if for a given fire class, the extinguisher type is correct...

{
 \"class A\": {
   \"ABC\": {
     \"correct\": true,
     \"explanation\": \"The extinguishing agent melts over the elements.\
  },
[...]

# Common Errors
Errors:
- Failure to check the fire extinguisher pressure gauge before use
  Action that leads to error: In cases where the fire extinguisher has a presure...
  Why it's wrong: If the extinguisher doesn't have pressure, you won't be able to...
[...]

# Navigation Aids
Aids to navigation:
Scenario 1
- Situation: The user does not remember how to move using teleport.
- How to detect it: At the beginning of the exercise, the user has not yet...
- What to tell the user: To move, you must press or move the joystick of your...
[...]
```

Figure 6: Game Context Prompt

B.3 Game Context + State Variables Prompt

Instructions

```
You are a helpful virtual trainer guiding a user through a virtual reality...
```

Game Context

```
# Procedural Steps
Below are the steps of the procedure included in the simulation with their...
[...]

# Extinguisher and Fire Type Incompatibility Errors
These errors check if for a given fire class, the extinguisher type is correct...
[...]

# Common Errors
Errors:
[...]

# Navigation Aids
Aids to navigation:
[...]
```

State Variables

```
{\"Action\": [{
        \"code\": \"-\",
        \"description\": \"Check extinguisher pressure gauge\",
        \"nature\": \"Action\",
        \"current_value\": \"No\"},
[...] ]}

\"Context\": [{
        \"code\": \"-\",
        \"description\": \"Available extinguishing agents\",
        \"nature\": \"Context\",
        \"current_value\": \"Water\"},
[...] ]}
```

Figure 7: Game Context + State Variables Prompt