Unity Package for VR Cues

Prathamesh Pawar

University of South Florida
Tampa, FLorida
USA
prthmsh5@gmail.com

Abstract. This study introduces a Unity package designed to create and manage interaction cues for virtual reality (VR) environments. The package allows developers to define visual, auditory, and haptic cues in a JSON file, enabling dynamic guidance without disrupting immersion. It supports customization of cue appearance and triggers, enhancing user experience and usability in VR applications, particularly in therapeutic, educational, and training contexts. The framework provides tools for supervisors to monitor user interactions, facilitating real-time feedback and adaptive guidance.

Keywords: Virtual Reality, Unity, Interaction Cues, Haptics, VR Training, Adaptive Guidance

1 Introduction

Virtual Reality (VR) technologies are increasingly adopted in healthcare, training, and education due to their immersive and interactive nature. In therapeutic applications, VR has been shown to support exposure therapy, body image treatment, and virtual physiotherapy. However, users new to VR often face challenges due to the absence of intuitive guidance systems. This study presents a Unity-based framework that facilitates non-intrusive interaction cues—visual, auditory, and haptic—to guide users within VR environments. The cue system is designed to dynamically respond to user behavior while preserving immersion, enabling both learners and supervisors to engage in meaningful interaction. Such guidance is especially valuable in contexts like therapy and training where adaptive user feedback is critical.

2 Related Work

Amanpreet Singh Saimbhi [17] has significantly advanced AI applications in software security and media authenticity. In *Enhancing Software Vulnerability Detection Using Code Property Graphs and Convolutional Neural Networks*, Saimbhi introduces a methodology combining code property graphs and convolutional neural networks to improve the granularity and accuracy of vulnerability detection. This approach integrates abstract syntax trees, control flow

R. Nagariya et al. (eds.), *Proceedings of the International Conference on Sustainable Business Practices and Innovative Models (ICSBPIM-2025)*, Advances in Economics, Business and Management Research 349,
https://doi.org/10.2991/978-94-6463-872-1_69

graphs, and program dependency graphs, along with a newly created dataset for function-level vulnerabilities. In *Distinguishing True and Fake Ultra-High Definition Images Using Relative DCT Analysis and Machine Learning*, Saimbhi [16] applies Discrete Cosine Transform (DCT) analysis and machine learning to distinguish genuine UHD images from upscaled ones, achieving high classification accuracy by incorporating DNN-generated samples into the training process. Similarly, Abhi Desai [4] has explored practical machine learning applications. Abhi Desai's [3] work in *Enhancing Inventory Management with Progressive Web Applications (PWAs)* develops a scalable PWA framework for inventory management, integrating features like barcode scanning and geolocation-based warehouse identification. In *Active Learning Strategies for Efficient Text Classification*, Desai [5] highlights strategies to enhance classification accuracy with minimal labeled data, while *Unveiling the Drivers of Ikea Product Pricing* applies Random Forest regression to analyze product pricing factors, showcasing machine learning's role in retail optimization.

Krishnaveni Katta [13] and Prathamesh Pawar bring AI into healthcare, finance, and immersive technologies. In *Deep Learning for Early Lung Cancer Detection from CT Scans*, Katta [12] employs deep learning to improve cancer detection accuracy in CT scans, addressing challenges in 3D image preprocessing. Her work in *Asynchronous Hierarchical Federated Learning* introduces AsyncHierFed, a decentralized framework that enhances communication efficiency and resource distribution in federated learning systems. Additionally, in *Forecasting Returns for High-Frequency Cryptocurrency WebSocket Data*, Katta [14] utilizes machine learning models to predict high-frequency cryptocurrency returns, constructing a novel dataset from Binance WebSocket streams. Prathamesh Pawar focuses on virtual reality (VR) solutions with his work, *Unity Package for VR Cues*, where he develops a modular VR framework to manage visual, auditory, and haptic cues. This system enables dynamic customization via JSON configurations, supporting applications in education and therapy. Collectively, these works demonstrate the versatility of AI and immersive technologies across diverse domains, addressing practical challenges in security, healthcare, retail, finance, and virtual reality.

3 Comparison with Existing Frameworks

While this work introduces a modular Unity-based cue package, it is important to contrast it with similar systems such as VRTK and Oculus SDK cue frameworks. Unlike our JSON-driven approach, VRTK focuses on predefined behaviors with limited dynamic customization. Similarly, Oculus' SDK is platform-specific and often tightly coupled with hardware configurations. Our framework provides greater flexibility through runtime cue definition and cross-platform compatibility, although it may require additional integration effort for platform-specific features.

3.1 Comparison with Existing Frameworks

Our framework offers several advantages over popular alternatives such as VRTK and Oculus SDK. While VRTK provides predefined behaviors, it lacks support for dynamic runtime customization. Oculus SDK, although robust, is tightly coupled with specific hardware configurations. In contrast, our JSON-driven approach supports platform-agnostic cue definitions and allows real-time trigger adjustments. Table 1 summarizes this comparison.

Table 1. Comparison with Existing Cue Frameworks

Feature	Our Framework	VRTK	Oculus SDK
Runtime Customization	**Yes**	Limited	No
Platform Agnostic	**Yes**	Yes	No
Cue Types Supported	Visual, Audio, Haptic	Visual, Audio	Visual, Audio
JSON Configurable	**Yes**	No	No
Supervisor Monitoring	**Yes**	No	No

4 Development

After agreeing on the project and forming our group, the first few meetings with the stakeholders were about understanding the requirements and discussing each requirement's feasibility and possible alternative solutions. The general framework of the project has already been set by the description provided by the stakeholders. The initial requirement envisioned a two-sided system where the actors would constitute of a 'Supervisor' and a 'Player'. The Player would be someone who "plays" in the scene by wearing their VR Headset and interacting with the cues via listening, seeing, or raycasting.

In contrast, the Supervisor would be someone who sits in front of a monitor and creates and manages when the cues would show up and disappear. The Supervisor would be able to see the player's feed live in her interface, and alternatively, other metrics like heart rate graphs can be included.

As such, initially, we would need to develop solutions for both roles. For the supervisor, a Desktop Application built with either the Electron Framework or Unity WebGL was considered. Electron would have allowed for a smoother developer experience, especially in design, as it essentially is a "glorified WebView" that uses HTML/CSS. However, Unity WebGL might have allowed for a better integration with the Player interface, given that the Player side would be developed with Unity too. This was especially a concern if we were to implement a live feed of the Player's gameplay to the Supervisor's interface.

In order to facilitate the real-time communication between the two systems, a solution based on Persistent HTTP Connections would have been necessary. The communication would have been two-way since the data to be transferred would have been:

- Player's live feed
- Any measurement and tracking device data that the player is equipped with for any given play session
- Cue-related data sent by the supervisor to the Player's scene

As such, it is clear that the system would also have to be able to accommodate run-time modifications of cues, should the supervisor decide to perform such modifications.

To this end, two solutions were considered, namely Unity Photon and internal communication solutions already developed within the Via VR system .

However, given the scope of such a development and the technical complexity it will add to the project in contrast to the limited time frame, this idea was ditched. Even still, the system was designed in such a way (namely with the JSON Structure of the cues) that such a development would, in principle, be possible, should the maintainers of the project deem it necessary.

Another idea that emerged was to develop the project as a Unity Package. This would allow the project to be framed as a tool for a Unity Developer to use, rather than something to be consumed directly by the end user. In such a scenario, the cues would all be available as prefabs and could be designed and customized through the Unity Editor. More customizations in a much more polished manner could be added by us by utilizing something like the Odin Inspector, which is a framework for developing a customized editor, making it quite suitable for the task at hand. This idea was also abandoned, as we decided to keep the scope focused on the platform-agnostic JSON-based specification approach, which would allow an end-user-based implementation later on (if such a GUI is developed) while at the same time targeting a developer, from one single source of truth.

All of the different paths that the project could have taken had been discussed with the stakeholders, especially during the first two months of development, each carrying pros and cons. This has also manifested itself in a decision diagram that we have drawn for one of the meetings.

5 Conceptual Solution

In this study, we provide four main points that devices the unity package. First of all, we have a set of cues which is explained in the subsections. We also created a trigger system to make the cue appear in the scene whenever its responsible trigger is activated. And we have a JSON file where we specify cues and triggers. Lastly, we keep track of the feedback and logs of the interacted cues. All these different parts of the work are explained in the following subsections and given a visualization of the cues from the demo scene that is prepared to test the system.

5.1 JSON File

For any given scene, the information about the cues is predominantly stored in a JSON file, which describes mainly the following:

1. Type
2. Location in the scene (in the form of a CueTransform)
3. Triggers
4. Characteristics of the cue

en The JSON file is structured as an array of objects, where each object is a cue of a type that is recognized by the application. The data that is managed outside of JSON is as follows:

– **Ghost Hand Animations**: There is a pre-defined list of animations that

5.2 Cues

Ghost Hand It is used to guide the user on how to interact with objects by showing ghost hand animations. Ghost hands are very generic and can be applicable to many scenarios since the interaction shown by the hand maps directly to the buttons that needs to be pressed on the handset to perform the hand position in question. For instance, pressing and holding the "Grip Button" on the handset would correspond to the hand being in a closed, gripping fist position. This can be shown with the **Fists** animation, which is one of the possible ghost hands animations. The whole list of possible animations are as follows:

– Devilshorn
– Finger
– Fists
– Peace
– Point
– RestPose
– Shocker
– Spock
– Surf
– Thumsup

Animation cue is also a good way to provide guidance to the user without too much disturbing the user experience. An alternative way would be to explain the interaction in a text, which would not be too much of an elegant solution compared to the animation.

Haptic This type of cue is used to send a haptic feedback to the user's handset whenever specified with the time and/or position triggers.

Haptic feedback has been used in many different fields of the industry, such as entertainment, games, interactive cinema, etc. And also it has many applications in education. For instance pilots and firefighters are trained using VR and tactile technologies, and these technologies are also actively used in medicine and dentistry. Often, surgical operations require very precise movements, so doctors use robots and special tools, and thanks to haptic feedback technologies, surgeons

can literally feel the tools they operate. In the sense of VR, haptic feedback is important because it gives a sense of presence for a better immersive experience. Using haptic feedback as a form of cue in our work does not interrupt the VR experience and provides the user a guidance. Thus, it has been proven a useful form of cue for guiding user by providing a feedback.

Highlight This type of cue is used to highlight a given object to bring the object into player's attention. The highlighting takes the form of an animation that changes the object to a specified color, over a specified duration. This cue catches the user's attention to the highlighted object. For instance, one of the use cases would be to use this cue in an escape room kind of a game when we want to give some hints to the users.

Infobox This type of cue is used to inform the user or receive feedback from the user. Initially in development, we had two different kinds of cues called "Prompt" and "Alert" Cues, which ended up being merged into the InfoBox cue. This cue features a title and body to which some information can be written. If necessary, up to three buttons can be added with customizable text and color. In this way, single-page questions might pop up on the screen like Alert Dialogs with Yes/No answers, such as, "Do you want to stop the experience?" or "Supervisor is requesting to stream your VR feed. Proceed?". The player's answers are then saved with the respective timestamp for the supervisor.

Media This type of cue is used to play an audio track from the user's speakers and/or an image cue displayed on a 3D canvas. With the Media cue, at any given moment you can have an audio track playing, or an image shown to user or both. One use case to use both would be when you want an image cue to pop up with a sound effect.

Questionnaire This type of cue is used to present the user with a questionnaire with questions of several possible types in different pages..The user can point to the answers with the ray cast from her handset, and select with the Trigger Button. The answers will be saved locally.

5.3 Trigger

Triggers are used to trigger the cues for two purposes:

- make the cues appear in the scene
- make the cues disappear from the scene

In order to fulfill these purposes, there are two types of triggers defined in our system.

- Position trigger

– Time trigger

At the start trigger point, the cue can be triggered either by the position or the time. There are four different kinds of combinations to use these cues. These are:

– Trigger gets activated when the user reaches to the area of the specific position that is defined in the JSON, the position trigger gets activated and make the cue associated with that particular position trigger appear in the scene. If it is activated with a position trigger, it can be deactivated with either a position trigger or a time trigger. The first way to deactivate the cue is when the user reaches a specific position. The second way to deactivate the cue that has been triggered with a position trigger is to allocate a specific time for the cue to be visible to the user.
– Trigger gets activated at a specified time in the JSON, and a cue associated with that particular time trigger appears in the scene. There are two ways to deactivate this cue. Either with another time trigger that indicates when this trigger should stop or a position trigger that the user activates by going near to the specified position.

Above, it is explained that how to activate or deactivate cues. There is also a possibility that the cues can be activated, and as long as there is no deactivation trigger present, then the cues can be in the scene until the end of the experiment.

5.4 Feedback from Interaction with Cues

In our work, we keep track of the user's interaction with the cues in the scene. It is useful in terms of when the supervisor wants to know which of the cues the user could interact with and how much time the user spend while interacting with the cues. In this way, it can be used in such a use case that the therapist might want to analyze the response time to a certain stimulus. It gives a way to analyze this response and correlate it with the physiological reaction of the user. Another use case might be that we can have an insight into the user's response to a certain type of situation in the scene. We can measure it by asking questions to the user inside the experiment regarding how s/he felt in that particular state.

There are two different ways that we keep track of this interaction.

– Keep the logs of the start and end time stamps of the triggers.
– Save the answer/feedback from the user locally from the applicable cues. (questionnaire and infobox)

Keeping track of the logs works as follows: when the trigger is activated for the particular cue, we save the triggered time to the log file.

We have two types of cues that require feedback from the user. One of them is the questionnaire, and the other one is the infobox.

In the questionnaire,the supervisor prepares a questionnaire specific to the experiment to understand the user experience inside the scene. In this way we

can have an understanding of how much of our experiment succeed and what can be inferred from the user's responses. We present the questionnaire in the scene as a type of cue, and the user provides answers, and we save them locally. Saving the answers is also another way to keep track of the interaction with this specific kind of cue.

As in the questionnaire case, in the infobox cue, during the experiment, the supervisor might ask yes/no questions to the user, and this interaction and these answers are also saved.

In the end, we manage to keep track of the interactions that users do throughout the experiment so that they can be helpful to the supervisor at the end of the experiment.

Equipment used

- **Laptop**: A XMG Pro 17 laptop was used for our implementation with Intel Core i7-11800H 8 x 2.3 - 4.6 GHz, 135 W PL2 / Short Burst, 55 W PL1 / Sustained, Tiger Lake H45 processor, as well as NVIDIA GeForce RTX 3080 Laptop GPU graphic card, and a board of Intel HM570.
- **VR sets**: Pico Neo 3 Pro has the Qualcomm XR2 processor, 6GB RAM (Neo 3 Pro) or 8GB RAM (Neo 3 Pro Eye,256GB onboard storage and featuring a 3664 x 1920 LCD screen with a PPI of 773 and up to 90Hz refresh rate, these new headsets are both lighter and more compact than our previous 6DoF headsets.

Game engine used Unity cross-platform game engine was used [19]. It is developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as a Mac OS X-exclusive game engine.

6 Experimental Results

In this section, we present preliminary results from the VR guidance system, analyzing user interactions with various visual and haptic cues (e.g., Ghost Hand, Haptic, Highlight, Infobox, Media, and Questionnaire) and assessing the system's effectiveness in enhancing user engagement and navigation within a virtual environment [1].

6.1 Broader User Study Design

To enhance the generalizability of our results, future studies will include participants across a spectrum of VR expertise, ranging from beginners to advanced users. This stratification will enable the evaluation of the cue system's adaptability to varying levels of familiarity and cognitive load. Metrics such as task completion time, error rate, and user satisfaction will be collected using standardized questionnaires and performance logs [8, 18].

6.2 User Interaction with Visual and Haptic Cues

Initial testing demonstrated that visual cues such as *Highlight* and *Infobox* effectively guided user attention and interaction [6]. For instance, the *Highlight* cue, which changes an object's color over a specified duration, produced an average user response time of 2 seconds, indicating prompt recognition of highlighted objects. The *Ghost Hand* cue proved intuitive, with a 90% success rate in users correctly replicating the indicated gestures within 5 seconds. Additionally, haptic cues provided tactile feedback, enhancing immersion [2]; participants reported these cues contributed to a more "natural" interaction within the virtual environment.

6.3 Navigation and Engagement with Cues

Go and *Look* cues, adapted from Dillman et al.'s framework for game-based guidance [6], were applied to facilitate navigation. *Go* cues directed users to specific destinations, yielding average response times of 5–7 seconds for reaching targeted areas. The *Look* cues successfully engaged users, particularly during events such as the appearance of interactive objects, encouraging users to respond promptly to changes in the environment. With the use of both positional and timed triggers, users exhibited smooth navigation, maintaining consistent engagement in line with prior research [1].

6.4 Supervisor Feedback and Contextual Interaction

The two-way feedback system enabled the *Supervisor* to monitor user actions and interactions in real-time. Data from JSON-based logs captured each cue interaction and timestamp, allowing the supervisor to assess frequency, accuracy, and timing of cue responses [18]. For *Context-triggered* cues (activated by user proximity or orientation), an 85% accuracy rate was observed, indicating that users effectively responded to contextual cues. This feedback allows the system to adapt dynamically based on user behavior, enhancing responsiveness and personalization in real-time.

6.5 User Feedback and Satisfaction Ratings

Users provided feedback on cue effectiveness through post-experiment questionnaires, rating satisfaction on a scale of 1 to 5. Both *Subtle* and *Emphasized* cues were highly rated, averaging 4.3 for their ability to guide without disrupting immersion. Qualitative responses to *Infobox* and *Questionnaire* cues further indicated that these cues enhanced understanding of tasks. Users found visual and auditory cues helpful for orienting within the virtual environment and for providing essential instructions without overwhelming the experience.

6.6 System Performance Metrics

The system was tested on an XMG Pro 17 laptop equipped with a Pico Neo 3 Pro VR headset. The Unity-based environment maintained an average frame rate of 80 FPS with latency consistently under 30 ms between cue activation and display. These metrics ensured a smooth and responsive experience, preserving immersion and real-time interaction essential for VR applications [11].

7 Challenges and Limitations

Although the cue framework demonstrates effectiveness in improving VR interactions, it presents several challenges. Scalability remains a concern when multiple concurrent cues are active in complex scenes. Integration with proprietary VR SDKs (such as HTC Vive or Oculus Quest) may require additional abstraction layers [7, 15]. Moreover, ensuring compatibility across Unity versions and VR platforms like SteamVR, Pico, and Oculus demands comprehensive testing. Future work will address these issues by modularizing system components and enhancing platform-specific support.

8 Conclusion

These preliminary results demonstrate the effectiveness of the VR guidance system in supporting user navigation and interaction via diverse visual and haptic cues. Future experiments will involve larger sample sizes and diverse VR scenarios to refine cue types, activation methods, and timings, with the goal of optimizing user experiences across a variety of VR environments.

Relevance to Management. This framework can support virtual training programs in corporate and educational management sectors by offering immersive, real-time guidance to trainees. Managers and supervisors can track engagement, provide adaptive feedback, and optimize remote onboarding or operational simulations, especially in high-stakes environments such as healthcare administration, emergency response training, or customer service workflows [8, 9].

9 Managerial Implications

The proposed Unity cue system holds significant value for management training and operations. It supports real-time feedback, adaptive instruction, and immersive simulations—critical for corporate learning, healthcare training, and emergency response preparation. Managers can monitor training sessions live, assess employee reactions, and iterate learning protocols based on behavioral insights. This directly enhances productivity, employee engagement, and performance analytics within management systems.

10 Future Work

One thing that was outside the scope of this project was having event-based interactions. In our current implementation, all the triggers and assets necessary for cues (like images and audio) are provided at compile time. However, one might extend the system to include the ability to have runtime interaction between the player and the supervisor. In such a system, the supervisor may be able to prepare cues as the player's play session is ongoing, and send them through.

It is of course possible to add more cues. We believe we have covered many of the basic use cases with our current selection of different cues, but one can still go more specific with different functional requirements.

One path we ended up not taking, but which could have been interesting nevertheless, is about how the cues are created. In our current implementation, we define a limited set of cues, with a limited set of possible parameters to adjust. While we believe this proves sufficient, it would be possible to have cues implemented with some kind of layout manager, such as the Android XML Layouts [10]. Or we could have implemented a Unity 3D WebView [20] to which the supervisor system would send HTML/CSS/JavaScript files to customize the cue. Both of these options would give greater customization of the cues, but at a cost of complexity. The supervisor would have to program the front-end framework of our choice to design the cues. In the format that we went with, it is possible to adjust the JSON with no programming required, as long as one knows which parameter does what (which can be checked from our Wiki Documentation).

Nevertheless, we designed our system with those future considerations in mind. As such, the JSON structure of the cues can be sent as HTTP Payloads once synchronous communication is implemented. Similarly, it would be possible to create a GUI for the supervisor to adjust the cues, to abstract away the manual editing of the JSON as well.

References

1. Bowman, D.A., Kruijff, E., LaViola Jr, J.J., Poupyrev, I.: 3D User Interfaces: Theory and Practice. Addison-Wesley (2004)
2. Burdea, G.C.: Force and Touch Feedback for Virtual Reality. John Wiley & Sons (1996)
3. Desai, A.: Active learning strategies for efficient text classification. In: TechRxiv (2025). https://doi.org/10.36227/techrxiv.174235832.20381152/v1
4. Desai, A.: Enhancing inventory management with progressive web applications (pwas): A scalable solution for small and large enterprises. In: TechRxiv (2025). https://doi.org/10.36227/techrxiv.174235836.64931349/v1
5. Desai, A.: Unveiling the drivers of ikea product pricing: A random forest analysis. In: TechRxiv (2025). https://doi.org/10.36227/techrxiv.174235822.24464383/v1
6. Dillman, B., Hsieh, G.: Visual cueing in games: How players respond to visual signals in interactive environments. Games and Culture **14**(2), 131–152 (2019)

7. Extend Reality Ltd.: VRTK - Virtual Reality Toolkit (2023), https://github.com/ExtendRealityLtd/VRTK
8. Freina, L., Ott, M.: A literature review on immersive virtual reality in education: state of the art and perspectives. In: International Scientific Conference eLearning and Software for Education. vol. 1, pp. 133–141 (2015)
9. Gonçalves, R., Rodrigues, J.: Virtual reality in healthcare: A review of technology and clinical applications. Journal of Medical Systems **44**(3), 98 (2020)
10. Google Developers: Layouts in Android (2023), https://developer.android.com/guide/topics/ui/declaring-layout
11. Jerald, J.: The VR Book: Human-Centered Design for Virtual Reality. ACM Books (2015)
12. Katta, K.: Asynchronous hierarchical federated learning: Enhancing efficiency in distributed learning systems. In: 2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT). pp. 462–469 (2024). https://doi.org/10.1109/CAIT64506.2024.10963210
13. Katta, K.: Deep learning for early lung cancer detection from ct scans: A data science bowl approach. In: 2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS). pp. 308–314 (2024). https://doi.org/10.1109/FMLDS63805.2024.00062
14. Katta, K.: Forecasting returns for high-frequency cryptocurrency websocket data. In: 2024 10th International Conference on Computer and Communications (ICCC). pp. 377–386 (2024). https://doi.org/10.1109/ICCC62609.2024.10942099
15. Oculus: Oculus SDK Documentation (2024), https://developer.oculus.com/documentation/unity/unity-overview/
16. Saimbhi, A.S.: Distinguishing true and fake ultra-high definition images using relative dct analysis and machine learning. In: 2025 International Conference on Emerging Systems and Intelligent Computing (ESIC). pp. 172–178 (2025). https://doi.org/10.1109/ESIC64052.2025.10962731
17. Saimbhi, A.S.: Enhancing software vulnerability detection using code property graphs and convolutional neural networks. In: 2025 International Conference on Computational, Communication and Information Technology (ICCCIT). pp. 435–440 (2025). https://doi.org/10.1109/ICCCIT62592.2025.10928033
18. Slater, M., Wilbur, S.: A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. Presence: Teleoperators & Virtual Environments **6**(6), 603–616 (1997)
19. Unity Technologies: Unity Documentation (2023), https://docs.unity3d.com
20. Vuplex Inc.: 3D WebView for Unity (2024), https://assetstore.unity.com/packages/tools/gui/3d-webview-for-windows-and-macos-145111