Biomechanically Constrained Gaze Control for NPCs in Unity: A VHP-Based Approach with Context-Aware Target Prioritization

Guadalupe Prado Saldanha Ribeiro[1], **Maria Andréia Formico Rodrigues**[1]

[1]CCT & Programa de Pós-Graduação em Informática Aplicada (PPGIA) & GIRA Lab
Universidade de Fortaleza (Unifor)
Av. Washington Soares 1321 – 60.811-905 – Fortaleza – CE – Brazil

guadalupeprado@edu.unifor.br, mafr@unifor.br

Abstract. Introduction: *Gaze is a central non-verbal cue for communicating attention, intention, and social presence in virtual characters. In games, however, Non-Playable Characters gaze and aiming behavior often remain mechanically rigid, visually inconsistent, or weakly connected to scene context.* ***Objective:*** *This paper presents a Unity-based gaze-control extension for the Virtual Human Project (VHP), designed to improve the plausibility of NPC gaze through biomechanical constraints, binocular coherence, and context-aware target prioritization.* ***Methodology:*** *The proposed approach modifies the VHP gaze pipeline with three mechanisms: angular eye clamping based on human ocular ranges, a unified focal-center strategy that prevents asymmetric eye convergence, and a target-weighting layer that increases gaze priority for semantically relevant objects. The system was implemented in Unity and analyzed through controlled runtime cases involving extreme gaze targets and multiple competing points of interest.* ***Results:*** *Runtime traces and visual inspections show that the extension limits excessive yaw and pitch, avoids cross-eye artifacts from independent per-eye constraints, supports head-eye coordination, and biases target selection toward user-defined contextual-priority objects without replacing the VHP gaze pipeline.*
Keywords *NPC Gaze, Procedural Animation, Virtual Humans, Unity, Eye-Head Coordination, Context-Aware Target Selection, Game AI.*

1. Introduction

Gaze is a key component of non-verbal communication. In virtual characters, it helps convey attention, intention, awareness, and social engagement [Garau et al. 2003, Slater et al. 2010]. In digital games, Non-Player Characters (NPCs) that look coherently at the player, threats, companions, or relevant objects tend to appear more attentive and believable [Luz Junior et al. 2021]. Conversely, rigid eye motion, excessive convergence, or gaze targets that ignore the scene context can reduce character credibility and contribute to unnatural behavior [Ruhland et al. 2014, Lee et al. 2002].

Despite the availability of procedural gaze tools and virtual-human animation frameworks, gaze control in game NPCs remains difficult to tune [Gorisse 2020, Ruhland et al. 2014, Luz Junior et al. 2025]. Developers must balance anatomical plausibility, binocular coherence, authorial control, runtime cost, and scene-dependent target selection. Rule-based systems are lightweight but may become rigid when stimuli compete for attention, while learning-based models require training data, reward design, and integration effort that may be unsuitable for small or medium-sized productions [Han and Lee 2022, Kim and Jung 2021, Lee 2024].

This paper investigates a practical middle ground: extending an existing Unity-based virtual human toolkit with explicit biomechanical constraints and context-aware target weighting. The work is grounded in the Virtual Human Project (VHP), an open-source package for procedural virtual-human animation in Unity [Gorisse 2020]. Rather than proposing a new gaze model from scratch, we improve the controllability and robustness of the existing gaze pipeline by adding: (i) anatomically motivated eye-angle limits based on reported ocular movement ranges [Lee et al. 2019]; (ii) a unified focal-center correction to avoid per-eye divergence artifacts; and (iii) a lightweight target-priority mechanism for semantically relevant objects. The study is guided by three research questions:

- **RQ1:** How can biomechanical constraints be integrated into an NPC gaze system without introducing cross-eye or excessive-convergence artifacts?
- **RQ2:** How can contextual object relevance be incorporated into procedural gaze target selection while preserving authorial control?
- **RQ3:** What are the practical benefits and limitations of this VHP-based procedural approach for game NPC gaze control?

The contribution of this paper is not a new machine-learning model. Instead, it presents a technically grounded extension of an existing gaze framework for game development. The main contributions are: (i) a VHP-based gaze-control architecture for NPCs in Unity; (ii) a biomechanical clamping mechanism for limiting eye rotations; (iii) a unified focal-center formulation that preserves binocular coherence; (iv) a context-aware target-weighting layer for prioritizing semantically relevant objects; and (v) a technical evaluation based on controlled runtime cases and debug traces.

2. Related Work

This work is related to human gaze behavior, gaze animation for virtual agents, procedural gaze tools, and learning-based decision models for NPC behavior.

2.1. Human Gaze and Virtual-Agent Plausibility

Human eye motion is constrained by anatomical limits and shaped by social context. Lee et al. report empirical ranges for ocular ductions, including horizontal and vertical asymmetries that are relevant for calibrating plausible virtual gaze limits [Lee et al. 2019]. Beyond static ranges, human gaze follows dynamic patterns of saccades and fixations, which support information acquisition and social communication. In virtual environments, gaze also affects presence and perceived communication quality. Garau et al. [2003] showed that gaze behavior interacts with avatar appearance and can influence the perceived quality of social interaction. Related work on virtual reality further shows that eye contact and gaze behavior can support collaboration even when avatar realism is imperfect [Slater et al. 2010].

These findings motivate gaze systems that combine mesh-level correctness, behavioral plausibility, and lightweight control. For games, gaze should remain constrained, coherent, and contextually meaningful without becoming costly or difficult to tune [Garau et al. 2003, Ruhland et al. 2014].

2.2. Procedural Gaze and Eye-Head Coordination

Procedural gaze models for virtual characters typically combine target selection, eye rotation, head motion, blinking, and micro-saccades. Ruhland et al. [2014] provide a broad review of gaze animation for virtual agents and robots, covering low-level mechanisms such as saccades and vestibulo-ocular reflexes, as well as high-level attention and interaction behavior. Krejša et al. [2018] propose real-time head-eye controllers for virtual agents and report that temporal coordination and physical constraints are important for naturalness. Models such as Eyes Alive [Lee et al. 2002] also demonstrate that procedural parameters can affect perceived agency and trustworthiness.

Recent work has also emphasized multimodal gaze coordination. HOIGaze, for example, models eye, hand, and head coordination during human-object interaction, showing that gaze is influenced by the motion of hands and manipulated objects [Hu et al. 2025]. Such results indicate that NPC gaze cannot be reduced to a simple "look at nearest target" rule. Instead, attention should depend on body motion, task relevance, social signals, and context.

2.3. Tools for Gaze Animation in Game Engines

Game engines provide practical platforms for integrating gaze behavior into interactive applications. The VHP toolkit offers procedural virtual-human animation in Unity, including gaze modes, facial behavior, lip synchronization, and event-based transitions [Gorisse 2020]. Commercial tools such as Realistic Eye Movements (REM) automate coordinated eye, head, and eyelid motion, and provide mechanisms for random scanning, looking at players, or focusing predefined points of interest [Knabe 2024]. These tools reduce implementation effort, but still require calibration for specific characters, rigs, and gameplay contexts.

The present work follows this tool-oriented direction. Instead of replacing existing gaze systems, we extend a VHP-based pipeline with additional constraints and target-priority logic. This makes the approach more accessible for game-development workflows, where full behavioral learning may be unnecessary or too costly.

2.4. Learning-Based and Context-Aware NPC Behavior

Machine Learning (ML) approaches have been used to generate more adaptive NPC behavior. Unity ML-Agents enables training agents through reinforcement learning and imitation learning in simulated environments [Han and Lee 2022]. Reinforcement Learning (RL) has also been explored for target tracking and navigation in games, often outperforming handcrafted state machines in dynamic scenarios [Kim and Jung 2021, Lee 2024]. However, RL methods introduce additional complexity, including reward design, training time, and state representation.

For gaze behavior, learning-based models can predict gaze from multimodal cues such as audio, head motion, or scene context [Canales et al. 2023, Itti et al. 2006]. LLM-based approaches have also been proposed to infer attention targets from social and situational context [Hwang et al. 2025]. These methods are promising, but they may exceed the needs of lightweight NPC control in many game-development scenarios. Our work therefore uses a procedural and interpretable mechanism, while leaving learning-based target prediction for future extensions.

3. System Requirements and Design Rationale

The proposed extension was designed around five requirements derived from the literature and from practical issues observed in the VHP gaze pipeline. Table 1 summarizes these requirements.

Table 1. Design requirements for the proposed gaze-control extension.

Requirement	Problem addressed	Implemented mechanism
Biomechanical plausibility	Eyes should not rotate beyond plausible or mesh-safe ranges.	Angular clamping for yaw, elevation, and depression.
Binocular coherence	Independent per-eye limits can create cross-eye artifacts.	Unified focal-center correction.
Focal safety	Very close targets may cause excessive convergence.	Minimum focal distance for the reconstructed target.
Eye-head coordination	Extreme targets should involve head/neck motion, not only eye rotation.	Reduced comfort limits that encourage IK participation.
Contextual relevance	Important objects should attract attention even when not closest.	Contextual target weighting layer.

These requirements address both low-level visual plausibility and high-level target selection. Biomechanical limits and focal safety prevent excessive eye rotation and convergence artifacts, while the unified focal center preserves binocular coherence. Eye-head coordination avoids forcing all target tracking onto the eyes, and contextual relevance allows authored objects to influence attention beyond distance-based selection.

4. Proposed Gaze-Control Architecture

Figure 1 summarizes the proposed architecture. The system receives candidate gaze targets from the scene, optionally modifies their priority through contextual weighting, computes a shared gaze direction from a central point between the eyes, clamps this direction in head-local coordinates, reconstructs a safe virtual target, and applies the same corrected target to both eye bones. Head and neck Inverse Kinematics (IK) are allowed to compensate when the target exceeds the comfortable eye-only range.

4.1. Context-Aware Target Prioritization

This component is a lightweight context-aware target-weighting layer. In the original gaze-selection behavior, targets can be weighted by spatial factors such as distance. However, game scenes often contain objects that are semantically relevant even when they are not the closest objects. To simulate relevance, semantic gaze targets are defined in Unity through the *semanticallyRelevantObjects* list. For listed objects, gaze priority is updated as: $newWeight = (currentWeight + 50) \times priorityMultiplier$. The +50 prevents relevant distant/static objects from being ignored. The *priorityMultiplier* controls fixation strength; higher values make gaze selection more dominant. In the original VHP implementation, the gaze target is updated at a random interval between 2 and 4 seconds, or whenever the target list changes.

The component listens to the target-selection process and checks whether a detected object belongs to a developer-defined list of semantically relevant objects or

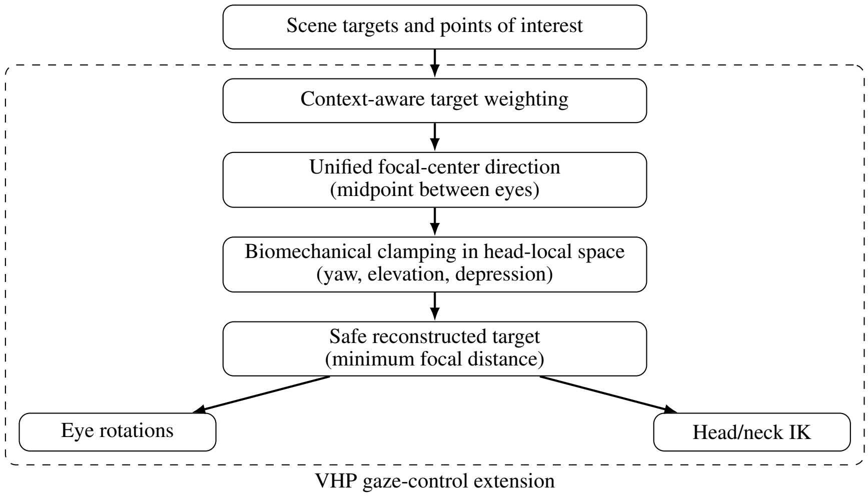

Figure 1. Overview of the proposed VHP gaze-control extension. The system combines contextual target weighting, a unified focal-center correction, biomechanical clamping, and safe target reconstruction before applying rotations to both eyes and allowing head/neck compensation.

high-priority objects. If the object belongs to this list, its gaze weight is increased by adding a base value of 50, ensuring that relevant static or distant objects are not ignored, and multiplying the result by a configurable multiplier in the range ([1, 10]); in this work, we used (5). This design does not replace the original probabilistic gaze behavior; it biases target selection toward semantically meaningful objects while preserving the existing VHP selection pipeline.

4.2. Unified Focal-Center Direction

A direct per-eye clamping strategy produced a failure mode: when each eye independently followed the original target, the difference between eye positions caused asymmetric angular saturation. One eye could reach the limit earlier than the other, producing cross-eye artifacts. To address this, the proposed approach computes gaze direction from the midpoint between the two eyes, here referred to as a unified focal center. This "cyclops eye" approximation produces a single direction vector that is clamped once and then converted into a shared virtual gaze target (a single 3D fixation point computed from a unified cyclopean gaze direction).

After clamping, both eyes rotate toward the same reconstructed target. This preserves binocular coherence and prevents each eye from solving a slightly different angular problem. The reconstructed target is also projected beyond a minimum focal distance, preventing excessive convergence for very close targets.

4.3. Biomechanical Clamping in Head-Local Space

This component restricts eye rotations according to configurable yaw and pitch limits. The initial calibration was informed by empirical ocular-motion ranges, including adduction, abduction, elevation, and depression values reported in the literature [Lee et al. 2019]. In practice, the values exposed to the Unity Inspector were refined to a more conservative comfort range, typically between 15° and 25°, to avoid mesh deformation and encourage the head/neck IK system to participate when targets are outside the comfortable eye-only range. The system exposes the main parameters in the Unity Inspector: maximum yaw, maximum elevation, maximum depression, minimum focal distance, and a debug option to lock head movement during tests. This makes the mechanism adjustable for different character rigs and meshes without changing the source code.

5. Implementation in Unity

The tool was integrated as a hybrid extension of Unity's VHP gaze pipeline. The native `VHPGaze` class was modified to add yaw/pitch limits and minimum focal distance, while `VHPGazeTarget` was extended with a weight-computation event. A decoupled `SemanticallyRelevantGazeManager` component subscribes to this event and adds semantic weights to predefined targets without replacing the native VHP selection logic. The central method is `ClampGazeDirection`, which computes a direction from the unified focal center, converts it to head-local coordinates, clamps yaw and pitch, and reconstructs a safe target in world space. Listing 1 summarizes the procedure.

Listing 1. Compact pseudocode of the unified focal-center clamping procedure.

```
centerEyePos = midpoint(leftEye.position, rightEye.position)
dirToTarget = normalize(target.position - centerEyePos)
localDir = inverse(headSpace) * dirToTarget

yaw   = atan2(localDir.x, localDir.z)
pitch = asin(localDir.y)

clampedYaw   = clamp(yaw,   -maxYaw,        maxYaw)
clampedPitch = clamp(pitch, -maxDepression, maxElevation)

clampedDir = headSpace * direction(clampedYaw, clampedPitch)
distance = max(distance(target, centerEyePos), minFocalDistance)
clampedTarget = centerEyePos + clampedDir * distance

leftEye.lookAt(clampedTarget)
rightEye.lookAt(clampedTarget)
```

The contextual weighting manager uses a Unity Inspector list of semantically relevant objects. If a candidate target belongs to this list, its weight is increased before selection, as summarized in Listing 2.

Listing 2. Compact pseudocode of contextual target weighting.

```
function updateTargetWeight(target, currentWeight):
    if target in semanticallyRelevantObjects:
        return (currentWeight + baseBoost) * priorityMultiplier
    return currentWeight
```

Figure 2 illustrates the Unity/VHP setup used to inspect contextual target selection. Speaker, anchor, and target icons denote audio cue, static contextual targets, and moving targets, respectively; colored cubes denote semantic-relevance targets. Figure 3 summarizes the main before–after cases used to inspect movement restrictions, focal-center correction, biomechanical clamping, and head-eye coordination. The Unity project source code is available at: `https://github.com/GuadalupePradoSR/extended-vhp`.

6. Technical Evaluation

The evaluation focuses on technical behavior rather than user perception. We analyzed controlled runtime cases designed to expose the main failure modes addressed by the implementation: excessive eye rotation, asymmetric convergence, target-selection

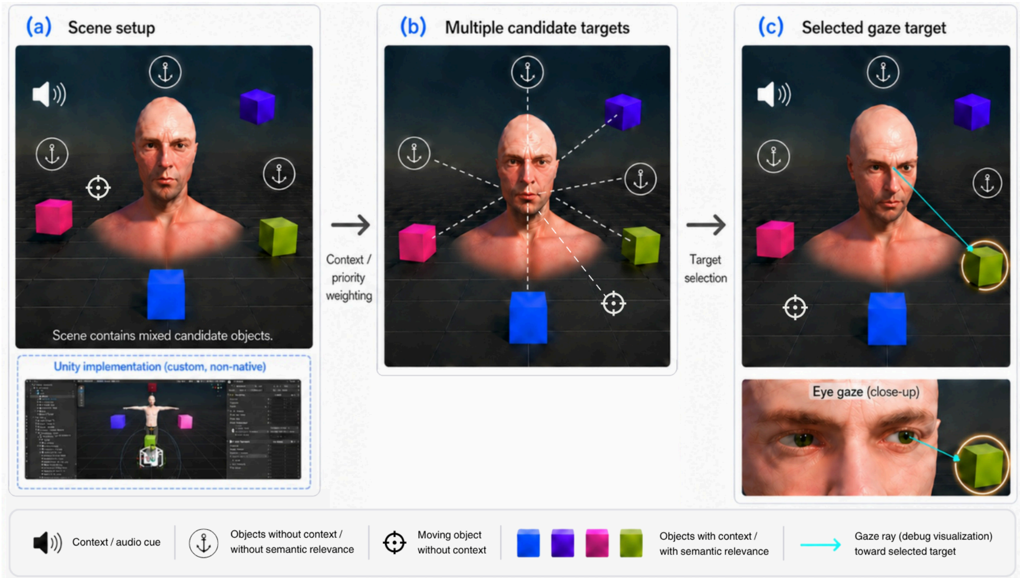

Figure 2. Gaze-control setup and target-selection test scene. The scene combines context-free candidate objects, contextual cues, moving objects, and high-priority semantic targets. The example illustrates how contextual weighting biases gaze toward the selected target while preserving the VHP procedural gaze pipeline.

ambiguity, and neck-stiffness artifacts caused by insufficient head-eye coordination. These cases were selected to verify whether the implementation behaves correctly under controlled failure conditions, not to measure perceived naturalness.

6.1. Evaluation Setup

The system was tested in Unity using VHP characters and scene targets placed at different positions relative to the avatar, including static, sound-emitting, moving, and high-priority contextual targets, as illustrated in Figure 2. Within the NPC's field of view, the tests considered: (i) extreme horizontal and vertical targets, to verify whether yaw and pitch were clamped; (ii) close targets, to verify whether the minimum focal distance prevented excessive convergence; (iii) off-axis targets, to inspect head-eye coordination and neck-stiffness artifacts; and (iv) scenes with multiple objects, to verify whether contextual weighting altered target priority as expected. The technical outputs were collected from Unity debug traces and visual inspection of the character. Table 2 summarizes the evaluation cases.

6.2. Clamping Behavior

The debug traces confirm that the clamping mechanism intercepts anatomically unsafe rotations. In one vertical case, the original pitch demand was $-33.48°$ and was clamped to $-15.00°$. In one horizontal case, the original yaw demand was $-20.27°$ and was clamped to $-15.00°$. These traces indicate that the method correctly limits extreme rotations in real time while preserving valid components of the original gaze direction. The unified focal-center strategy also addressed the induced-strabismus problem observed in the initial implementation. Because both eyes receive the same reconstructed virtual target, the final rotations remain visually coherent even when the original target lies outside the comfort

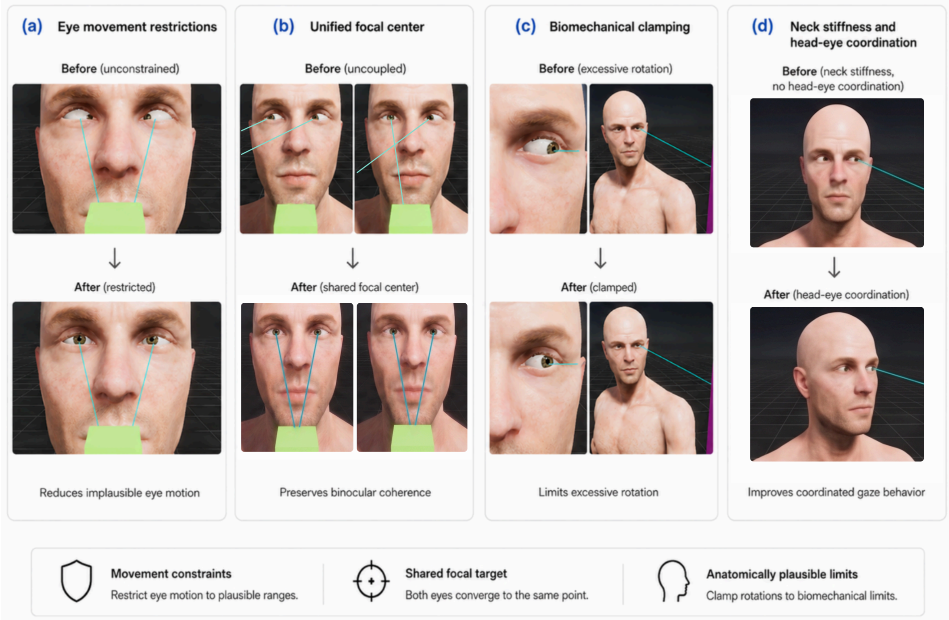

Figure 3. Representative gaze-control cases in the Unity/VHP prototype. Each column contrasts a problematic condition with its corrected outcome: (a) movement restrictions reduce implausible eye motion; (b) unified focal-center correction aligns both eyes to the same virtual target; (c) biomechanical clamping limits excessive eye rotation; and (d) head-eye coordination reduces neck-stiffness artifacts. Cyan lines show debug rays.

range. This is especially important for NPCs that must track fast-moving or off-axis objects while preserving facial plausibility.

6.3. Contextual Target Weighting

The contextual weighting mechanism changed target priority as expected. In the tested case, the selected high-priority object named `Cube` had its weight increased from 100 to 1,500, making it the dominant gaze candidate. The goal was not to exhaustively tune weight values, but to verify that semantic weighting could alter target-selection probabilities in a controlled way. Weights act probabilistically rather than deterministically: higher values increase selection likelihood, while lower-weight targets may still be briefly selected. Multiple important objects receive the same increment and compete according to final salience, including weight, position, distance, movement, and sound cues. Equal-weight targets occupy equivalent probability ranges and are selected through `Random.Range` when the target-switching timer expires. This result does not imply human-like emotional reasoning; rather, it shows that author-defined semantic relevance can be integrated into a procedural gaze-selection loop with minimal overhead and without discarding the original VHP behavior.

7. Results and Discussion

The extension satisfies the design requirements in Table 1 through the controlled cases in Table 2. It clamps unsafe rotations, reduces asymmetric convergence, and adds author-defined target relevance beyond distance-based attention. Table 3 confirms this behavior with runtime pitch/yaw traces restricted to the configured limits.

Table 2. Controlled runtime cases used in the technical evaluation.

Case	Condition	Expected behavior
Angular limits	Target requires excessive pitch or yaw.	Eye rotation is clamped to configured limits.
Close target	Target is too close to the eyes.	Reconstructed target respects a minimum focal distance.
Multiple objects	Several objects compete for attention.	Objects with semantic relevance receive higher target weight.
Head-eye coordination	Target is outside comfortable eye-only range.	Head/neck motion contributes to gaze, reducing neck-stiffness artifacts.

Table 3. Examples of runtime clamping traces.

Axis	Original	Clamped	Interpretation
Pitch	$-33.48°$	$-15.00°$	Excessive downward gaze was restricted to the configured limit.
Yaw	$-20.27°$	$-15.00°$	Excessive horizontal gaze was restricted to the configured limit.

The approach remains lightweight and authorable, since angular limits, focal distance, and priority multipliers are edited directly in Unity. Its limitations are that relevance is author-defined, the evaluation is technical rather than perceptual, and the system does not yet model full-body orientation, hand-object coordination, emotional state, or dialogue timing.

7.1. Threats to Validity

This study has limitations. The evaluation uses controlled technical cases and debug traces, so the results are implementation evidence rather than perceptual validation. Thresholds were calibrated for one VHP character and may require adjustment for other rigs or stylized meshes. The contextual-priority mechanism uses author-defined priorities and does not infer semantics automatically. Finally, no user study was conducted, so perceived naturalness, believability, and engagement remain future work.

8. Conclusion and Future Work

This paper presented a Unity/VHP extension for NPC gaze control that combines biomechanical eye clamping, unified focal-center correction, minimum focal-distance handling, head-eye coordination, and context-aware target weighting. The technical evaluation shows that the system constrains excessive yaw and pitch, avoids cross-eye artifacts from independent per-eye clamping, and biases gaze toward user-defined semantically relevant objects. The contribution is a practical, interpretable procedural mechanism for improving NPC gaze plausibility in Unity without learning-based training. Future work includes perceptual user studies comparing the extension with the original VHP behavior and rule-based baselines, as well as more sophisticated contextual models involving dialogue, object affordances, emotional state, character goals, and gameplay context.

Acknowledgments

The authors thank Universidade de Fortaleza, GIRA Lab, CNPq (Grants 172556/2025-2 and 314776/2023-0), and the reviewers for their valuable feedback.

References

Canales, R., Jain, E., and Jörg, S. (2023). Real-time conversational gaze synthesis for avatars. In *Proceedings of the 16th ACM SIGGRAPH Conference on MIG*, pages 1–7.

Garau, M., Slater, M., Vinayagamoorthy, V., Brogni, A., Steed, A., and Sasse, M. A. (2003). The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 529–536.

Gorisse, G. (2020). Virtual Human Project (VHP). `https://github.com/GeoffreyGorisse/VHProject`.

Han, S. H. and Lee, Y.-H. (2022). Implementation of Target Object Tracking Method using Unity ML-Agent Toolkit. *KSDET*, 21(3):110–113.

Hu, Z., Haeufle, D., Schmitt, S., and Bulling, A. (2025). HOIGaze: Gaze Estimation During Hand-Object Interactions in Extended Reality Exploiting Eye-Hand-Head Coordination. In *Proc. of the Special Interest Group on Computer Graphics and Interactive Techniques*, SIGGRAPH Conf. Papers'25, New York, NY, USA. ACM.

Hwang, J., Hong, S., Seon, J., and Kang, H. (2025). How Does a Virtual Agent Decide Where to Look? Symbolic Cognitive Reasoning for Embodied Head Rotation. In *Proc. of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12.

Itti, L., Dhavale, N., and Pighin, F. (2006). Photorealistic attention-based gaze animation. In *Proc. of the 2006 IEEE Intern. Conf. on Multimedia and Expo*, pages 521–524.

Kim, D. and Jung, H. (2021). Performance analysis of target tracking AI based on unity ML-Agents. *The Journal of Korean Inst. of Information Tech.*, 19(12):19–26.

Knabe, T. (2024). Unity Asset: Realistic Eye Movements. Last visited: 7 April 2026.

Lee, S. P., Badler, J. B., and Badler, N. I. (2002). Eyes Alive. In *Proc. of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 637–644.

Lee, W. J., Kim, J. H., Shin, Y. U., Hwang, S., and Lim, H. W. (2019). Differences in eye movement range based on age and gaze direction. *Eye*, 33(7):1145–1151.

Lee, Y.-H. (2024). Implementation of Intelligent Agent Based on Reinforcement Learning using Unity ML-Agents. *IJIBC*, 24(2):205–211.

Luz Junior, J. A., Ribeiro, G. P. S., Pessoa, R. F., Magalhães, A. H. T., and Rodrigues, M. A. F. (2025). Generative AI for facial expressions in 3D game characters: A retrieval-augmented approach. In *Proc. of the 9th ICSE GAS*, pages 9–16. IEEE/ACM.

Luz Junior, J. A., Rodrigues, M. A. F., and Hammer, J. (2021). A storytelling game to foster empathy and connect emotionally with breast cancer journeys. In *Proc. of the 9th Intern. Conf. on Serious Games and Applic. for Health (SeGAH)*, pages 1–8. IEEE.

Ruhland, K., Andrist, S., Badler, J., Peters, C., Badler, N., Gleicher, M., Mutlu, B., and Mcdonnell, R. (2014). Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems. In *Proc. of Eurographics*, pages 69–91, Strasbourg, France.

Slater, M., Spanlang, B., Sanchez-Vives, M. V., and Blanke, O. (2010). First person experience of body transfer in virtual reality. *PloS one*, 5(5):e10564.