A Unity-Based Framework for Sound Transmission and Perception in Video Games

Jorge Calado, Pedro Santana, and Rui J. Lopes
ISCTE - Instituto Universitário de Lisboa, Portugal
Instituto de Telecomunicações, Portugal
{jmpsc, pedro.santana, rui.lopes}@iscte-iul.pt

***Abstract*—The ability for humans to hear, attend to, and understand incoming sounds is affected, at least, by environmental, morphological, and cognitive factors. Conversely, current implementations of audition in virtual characters often only consider as constraint to the auditory process the distance between the sound emitter and the sound receiver. To cope with this limitation, this paper presents ongoing work on a novel framework integrated in Unity, directed to game developers interested in implementing non-player characters with noisy, character-specific, and context-dependent auditory perception.**

***Keywords*—computer games; auditory perception; non-player character behaviour; sound perception; senses; sound.**

I. Introduction

Humans rely on several senses to perceive the surrounding environment. Sound perception is a key feature for the perception of the surroundings, as it complements sight and enables listeners to perceive the environment 360 degrees around them. The perception of sound is key when it comes to the simulation of human behaviour, thus sound plays an increasingly important role in video games (Bostan, 2009).

As with sight, virtual characters rely on hearing to perceive the surrounding environment in a human-like manner. Usually, the way sound is perceived in video games only takes into consideration the distance between the emitter and receiver, disregarding all other properties of sound and human perception, which are important parameters of human-like hearing (Millington and Funge, 2016).

This paper introduces a framework for the cross-platform gaming engine Unity, which enables the developer to integrate transmission and perception of sound into the characters that are not directly controlled by the player, Non-Player Characters (NPC).

The paper is organised as follows. Before presenting the proposed framework and its integration in Unity in Section III, Section II discusses the way sound is currently considered and implemented in video games. Details regarding key aspects of human auditory perception exploited in the proposed framework are also analysed. Then, in Section IV, a custom game called Fortress will be presented as a case study to validate the proposed framework. Finally, some conclusions and future work directions are drawn in Section V.

II. Related Work

In the current generation of video games, in spite of the growth in processing power, sound perception in NPCs is still simplified to meet a sufficient degree of realism. Currently there are two predominant perception systems implemented in NPC: Region Sense Manager (RSM) and Finite Element Model Sense Manager (FEMSM) (Millington and Funge, 2016).

RSM provides a simplified implementation of sensory perception using spherical regions. When implementing RSM, spherical regions are defined around the emitter, each having different diameters. An attenuation to the sound's intensity is defined between each spherical region. When the listener is located inside the region between two of the spherical regions, the emitter calculates whether the listener perceives the sound, given the spherical region attenuation and the listener's acuity (see Fig. 1).

The FEMSM method divides the virtual environment into discrete spaces, each determining how senses should be handled. The usual approach is to create a senses graph, which describes the properties of each element in the discrete space. When a sound is emitted, it starts at the node where the emitter is located. The sound is then replicated to the neighbour nodes, and its intensity attenuated accordingly. If a listener is at a node with a sound of audible intensity, the sound is perceived (see Fig. 2).

Zambetta (2007) describes a model with resembles FEMSM but revolves around creating a graph which is navigated each time a listener needs to perceive the environment. When perceiving, a search is made for any audible sounds and fuzzy predicates are applied when evaluating whether a given stimuli is perceptible. Each character has a set of predicates in order to simulate different hearing capabilities. This offers some degree of limitation to the hearing capabilities of NPCs, providing a less perfect, but still rather artificial sound perception.

Both RSM and FEMSM fail by delivering a sound in a single frame, as a fast moving character might pass past the area of the sound when sound should linger in the air. This problem could be solved in FEMSM as the sound could stay in a node and rapidly decay, but this would consume a great amount of memory and processing power.

RSM is less demanding when it comes to processing power, but it fails to provide a way to calculate the interaction of sound with obstacles.In RSM sound is perceptible through walls and other objects that offer no resistance to the propagation of sound. This could be solved with the implementation of additional heuristics.

In addition, RSM and FEMSM approaches do not consider the psychological filtering performed by the human auditory cortex. In both methods, to determine whether a given sound has been heard by the NPC, its intensity is checked against a given threshold. However, sound must be characterised by many other features, rendering an intensity-based thresholding rather simplistic and, thus, an unrealistic simulation of the human auditory perception.

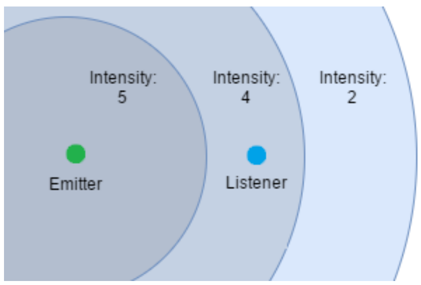

Figure 1: A two-dimensional view of the RSM spheres, each representing an area where the sound has a different intensity. The intensity of the sound is proportional to the diameter of each sphere.

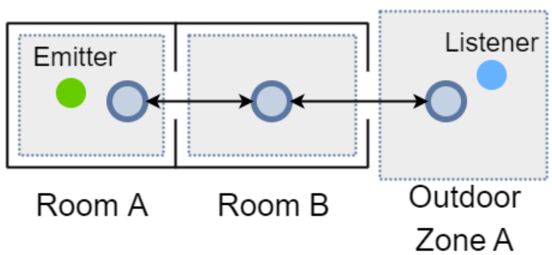

Figure 2: A FEMSM graph of a map with only three distinct zones, with each node representing a whole division's information of the sound transmission attenuation.

III. Proposed Framework

The transmission of sound to the NPC is just the beginning of the processes that must be modelled. The perception of sound should be affected by the NPC's current state, personality, and the action under execution, that is, the current context.

A. Human auditory perception

Soon after a sound reaches the human ear, the auditory cortex must determine whether the sound is relevant, given the current task (Fritz et al., 2007). Sounds which are not relevant are considered noise and, therefore, discarded from further processing.

The listener may also identify the source of the sound, which is a significant feature to consider in video games where NPCs and player share a relationship tree (Afonso and Prada, 2008). The use of a relationship tree would enable each NPC to possibly identify the identity of the emitter of a given sound, and if the emitter has a high priority relationship with the listener, the listener could give priority to that emitter's sounds. We all often observe this phenomenon in our daily lifes, such as when we feel higher hearing acuity towards the voice or cry of our children.

Not all humans have the same hearing acuity. With age, the range of frequencies a human can perceive gets narrower. This narrowing may be aggravated by hearing trauma caused by exposure to loud or continuous noises (Cruickshanks et al., 1998). In these cases, the sound reaches the ear but the auditory cortex is unable to process it. Genetics also take part when determining the human listening acuity.

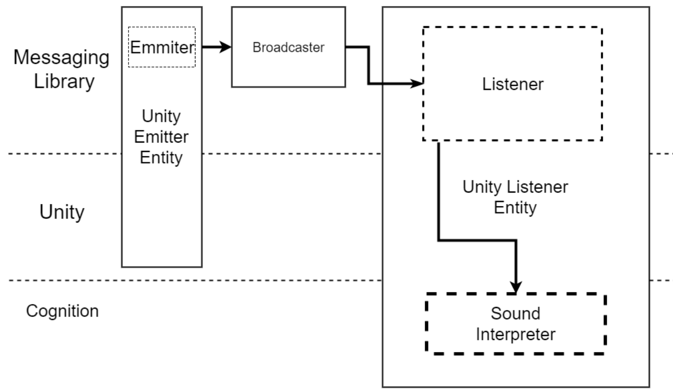

Figure 3: Diagram of the components' interaction during the transmission of a sound message in the proposed framework. The sound starts at the Unity Emitter Entity, is broadcast through the Messaging Library, and then delivered to the Unity Listener Entity.

B. Framework composition

The proposed framework is divided into three different layers (see Fig. 3):

1) **Messaging module** is responsible for the transmission of messages from the emitters to the listeners. The interface of the communication between the emitters and listeners is defined in this module, which also provides the structure that describes a sound message.
2) **Engine Integration Module** is the game engine which ties together the Messaging Module and the Cognition Module into the virtual world.
3) **Cognition Module** defines the behaviour of the NPCs, which receives the auditory stimuli information and, then, provides the set of actions to attain a given behavioural objective.

C. Sound Descriptor

A sound descriptor outlines the properties of the sound, such as intensity, frequency range, emitter id, emitter location, and speech transcript. These properties are used by the filters to evaluate if the sound is perceived by the Cognition Module so that the NPC can act based on the sound's description.

D. Emitter

Emitters send Sound Descriptors to the Listeners through a Broadcaster. The emitter is responsible for the creation of the

Sound Descriptor. An emitter might also broadcast a sound as noise. When a sound is emitted as noise, it is considered continuously active and might affect the way other sounds are perceived by the NPC.

E. Broadcaster

Broadcasters are responsible for the propagation of sound from the emitters to the listeners. A broadcaster adds the position of the emitter to the sound descriptor and, depending on the emitter, it flags the sound as regular sound or noise. When an emitter is created, it is automatically registered to the default Broadcaster or it might register to an alternative one, in case there is the need to simulate an alternate auditory communication channel. A listener may also be associated with a Broadcaster instance different from the default one.

F. Listener

Each listener is associated to a set of filters. Each filter describes a way of calculating the saliency of the sound message by analysing its parameters. If after the application of all the filters the message has a non-positive saliency, it is regarded as not perceived by the NPC and, thus, discarded. If a sound has a positive saliency, it is queued as audible, being afterwards processed as such. The sound is only processed if there is no concurrent sound within the same time-frame, thus stealing the listener's attention. If this happens, the sound is discarded as it is considered noise.

G. Filters

Filters are implemented within the game engine, providing a simple way to tweak its parameters inside the game authoring tool. They allow the developer to model human's hearing complexity by adapting the properties of the Sound Descriptors.

Filters are attached to the Listener during the creation of the latter. Moreover, filters may be tuned, added, or removed in run-time. Each filter sequentially analyses the sound properties to determine whether its overall saliency should be either increased or decreased.

In the current implementation, the following base filters were created:

1) **Distance Falloff filter** evaluates whether the sound is within the hearing range of the Listener, lowering the sound's saliency the further away the listener is from the source.
2) **Hearing Loss filter** checks whether the sound is actually perceptible according to the Listener's hearing health, given the sound's intensity and frequency. This is simplified to make the NPC's hearing lowered by a certain percentage.
3) **Relationship filter** verifies the kind of relationship between emitter and listener. The sound saliency is increased if the emitter has an important relationship with the listener.
4) **Noise filter** checks whether a sound can be heard by the Listener, given the noise level present in the environment. A sound is considered noise if it is emitted with the noise flag on.

Many more filters can be created and attributed to NPCs, with the price of processing power. Each filter should keep its operation as simple as possible, as it has to be computed at each game state update, and simultaneously, for several NPCs and multiple sound sources.

As soon as a sound is perceived, the message that it conveys can be processed by the Cognitive Module, so that the NPC can act upon the newly acquired stimuli.

H. Integration in Unity

The framework was integrated in the gaming engine Unity. Each element present in a Unity game scene is called an entity. Each entity is associated to a set of behaviours that are iterated in each game update cycle.

A virtual character entity may be associated to an emitter behaviour, a listener behaviour, or both. These behaviours were created to make the bridge between the Messaging Module and the game engine, enabling the synchronisation of the sound playback, as well as the sound descriptor broadcast and reception.

In the current implementation, each NPC entity is associated to a behaviour tree, composed by action nodes. The query for new messages is made by the base action node, as the listener must be receptive to auditory cues at every moment, despite the action it is currently performing. When the query delivers a message, the action node that receives it must act accordingly.

An entity can accumulate both Listener and Emitter roles, allowing it to emit and perceive sound. Sounds are broadcasted immediately once their are sent by an entity (asynchronously), but are only processed when the listening NPCs update cycle iterates (synchronously).

IV. Case Study

To test the integration of the framework in Unity, a simple First-Person-Shooter (FPS) game, Fortress, was created. The game takes place within the walls of a fortress and there are patrolling NPCs that walk around the fortress. When a NPC hears the player, the former rapidly proceeds to look for the source of the sound and, then chases it (the player). After being caught, the player must start from the beginning. The player's objective is to reach the door of the main building without being heard or seen by the NPCs. Being a FPS game, the camera shows the player's character first-person perspective (see Fig. 4).

As mentioned, to include the proposed sound perception framework, each Unity NPC entity must be associated to a Listener Behaviour, responsible for receiving all sound messages. Each NPC must also be associated to a Unity Behaviour corresponding to three different filters: Distance Falloff, Hearing Loss, and Noise. The noise filter affects the sound saliency based on the amount of noise surrounding the NPC. If there is too much noise, the incoming sound may not be loud enough to be heard by the NPC. For example, in the Fortess game, players could take advantage of the noise emitted by certain elements in the map (e.g., the sound of a

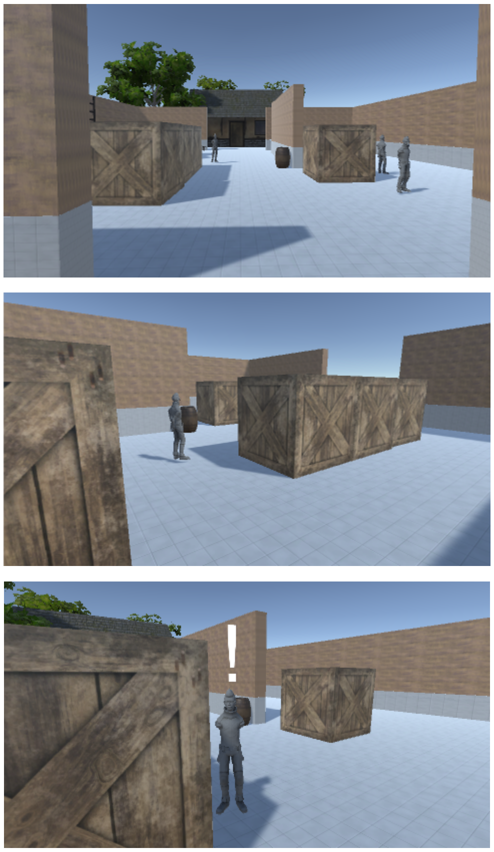

Figure 4: Screenshots of the Fortress game. Top and middle: two situations in which the set of NPCs are unable to hear the player's footsteps. The inability to detect the player is due to the sound falloff filter that implements distance-based sound attenuation. Bottom: situation in which a sound is perceived by the NPC. The exclamation mark represents the fact that the NPC has perceived the sound of the player's footsteps.

car engine or a radio playing music loud) to hide their acoustic signature and, thus, move stealthily through the map without being noticed.

For the case study, the Player entity was associated to an Emitter Behaviour so that the player's footsteps could be heard by the NPCs. Each element that emits noise in the environment (e.g., power generators and radios) were also associated to Emitter Behaviours. These notify the Listener Behaviour for the existence of noise. For the sake of the framework's validation, NPCs' visual perception was turned off. This ensures that only sound triggers the player's detection.

To assess the validity and usefulness of the proposal, we have tested NPCs from perfect hearing to faulty hearing. Initially, as in the RSM method (see Section II), all NPCs were set to have the same artificial hearing acuity, affected only by the distance to the sound source. With this approach the NPCs were unable to exhibit individual auditory traits. Then, all filters developed for the framework (not limited to distance attenuation) were turned on so as to attain the so desired imperfect, i.e., human-like, hearing. These filters were parameterised in order to tailor the hearing acuity of the characters according to the uniqueness of their desired personalities. As this case study only considers a few NPCs, it is not yet possible to truly assess the computational impact of the proposed framework.

V. Conclusion

A framework for inclusion of auditory perception in NPC was presented. The framework offers a flexible implementation of auditory perception based on a set of filters that mimic the human's physical and psychological mechanics responsible for the perception of sound. Preliminary tests suggest that the framework is able to offer more natural auditory perception to NPCs than previous methods. That is the case because the proposed framework allows authoring NPC with individual context-sensitive auditory traits. The following presents some limitations of the current system and points out possible improvement directions.

Computational performance of the framework may prove to be an issue when coordinating a large number of sound-aware NPCs. To cope with this issue, sound messages directed to NPCs that are not within range of the sound could be automatically discarded. The current implementation does not provide a visual authoring tool to customise the auditory trait of each NPC, that is, the parameters of its associated filters. Such a visual tool should also include the ability to switch parameter values according to different game contexts.

In the current implementation of the framework, when two or more sounds are received by a NPC on the same update cycle, the sound with the higher saliency is processed and the others are discarded. However, humans in many scenarios are able to handle more than one sound at a time, hence, future work is still required to better model the effects of concurrent sounds and background noise in NPC hearing.

References

Afonso, N. and Prada, R. (2008). Agents that relate: Improving the social believability of non-player characters in role-playing games. In *International Conference on Entertainment Computing*, pages 34–45. Springer.

Bostan, B. (2009). Player motivations: A psychological perspective. *Computers in Entertainment (CIE)*, 7(2):22.

Cruickshanks, K. J., Wiley, T. L., Tweed, T. S., Klein, B. E., Klein, R., Mares-Perlman, J. A., and Nondahl, D. M. (1998). Prevalence of hearing loss in older adults in beaver dam, wisconsin the epidemiology of hearing loss study. *American journal of epidemiology*, 148(9):879–886.

Fritz, J. B., Elhilali, M., David, S. V., and Shamma, S. A. (2007). Auditory attention—focusing the searchlight on sound. *Current opinion in neurobiology*, 17(4):437–455.

Millington, I. and Funge, J. (2016). *Artificial intelligence for games*. CRC Press.

Zambetta, F. (2007). Simulating sensory perception in 3d game characters. In *Proceedings of the 4th Australasian conference on Interactive entertainment*, page 27. RMIT University.