



Pushing Out the Classroom Walls: A Scalability Benchmark for a Virtual Audience Behaviour Model in Virtual Reality

Yann Glémarec, Jean-Luc Lugrin, Anne-Gwenn Bosser, Paul Cagniat, Cédric Buche, Marc Erich Latoschik

► To cite this version:

Yann Glémarec, Jean-Luc Lugrin, Anne-Gwenn Bosser, Paul Cagniat, Cédric Buche, et al.. Pushing Out the Classroom Walls: A Scalability Benchmark for a Virtual Audience Behaviour Model in Virtual Reality. Mensch und Computer, 2020, <10.18420/muc2020-ws134-337>. <hal-03429110>

HAL Id: hal-03429110

<https://hal.science/hal-03429110v1>

Submitted on 15 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Pushing Out the Classroom Walls: A Scalability Benchmark for a Virtual Audience Behaviour Model in Virtual Reality

Yann Glémarec

ENIB - Lab-STICC UMR CNRS 6285
Brest, France
HCI Group - Würzburg University
Würzburg, Germany
glemarec@enib.fr

Jean-Luc Lugin

HCI Group - Würzburg University
Würzburg, Germany

Anne-Gwenn Bosser

ENIB - Lab-STICC UMR CNRS 6285
Brest, France

Paul Cagniat

ENIB - Lab-STICC UMR CNRS 6285
Brest, France

Cédric Buche

ENIB - Lab-STICC UMR CNRS 6285
Brest, France

Marc Erich Latoschik

HCI Group - Würzburg University
Würzburg, Germany

ABSTRACT

In this paper, we describe the implementation and performance of a Virtual Audience behaviour model for Virtual Reality (VR). The model is a VR adaptation of an existing desktop model based on crowd-sourced data. The system allows a user in VR to easily build and experience a wide variety of atmospheres with small or large groups of virtual agents. The paper describes the model's implementation and possible benefits for different applications, e.g. training, therapeutic or culture. Our first scalability benchmark results demonstrated the ability to simultaneously handle one hundred virtual agents without significantly affecting the recommended frame rate for VR applications. Lastly, we propose a set of improvements our training system can benefit from. This research is conducted in the context of a classroom simulation software for teachers' training.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; • **Software and its engineering** → **Virtual worlds training simulations**.

KEYWORDS

Virtual Reality, Behaviour Model, Education, Virtual Agent

1 INTRODUCTION

A number of recent virtual reality (VR) training systems use groups of virtual agents reacting to the user. The virtual characters populating these virtual environments are called a Virtual Audience (VA). For instance, VR training systems simulating different VAs (e.g., attentive or bored) have been used to reduce public speaking anxiety [5, 9, 19]. Classroom simulations have also been used to help trainee teachers to manage students by reproducing non-verbal behaviours commonly met [7, 10, 16]. The term atmosphere

is used in this context to describe how different types of audiences may be perceived [7]. Studies have defined a set of virtual agents behaviours to simulate distinguishable audience styles [12], and further work identified a set of critical non-verbal behaviours influencing VA perceptions [4]. More recent research evaluated these kind of behaviours in VR in the context of teacher training simulations [6, 20]. These behaviours can be simple non-verbal signals (e.g. a head movement or a smile) or a combination of complex signals through time (e.g. gaze away time mixed with posture's proximity and openness and facial expressions).

However, easily creating a highly believable audience, composed of a number of virtual characters enacting specific desired atmospheres, is still a challenge. Having a system which allows the handling of large groups of agents without compromising the VR experience is the first step to take on this challenge. Indeed, numerous user studies demonstrated the negative impacts of high latencies, temporal jitter, and positional error on user performances, satisfaction, discomfort and sense of immersion [1, 8, 11, 13, 15, 17, 18, 21]. Consequently, it is important to first benchmark the system in order to be aware of how many agents the system can manage without increasing latency.

In this paper, we present a new open-source tool to quickly and easily design VA atmospheres in a generic way without requiring expertise in animation or computer graphics. In contrast with our system, and according to our knowledge, existing tools [2, 12] do not allow creating atmospheres interactively in VR, and have not been evaluated for large VA. Because our long term goal is to provide a virtual classroom training system allowing instructors to dynamically adapt the classroom to a teaching scenario we are first looking to the model implementation's limits and scalability performances in order to lay out a believable system following VR latency recommendations. Thus, we are here focusing on exposing our system's scalability performances as well as its architecture.

2 RELATED BEHAVIOUR MODEL

The behaviour model we implemented is from Mathieu Chollet and Stefan Scherer [4]. This particular model has demonstrated the capacity to express relevant audience states (e.g. bored, interested).

To do so they created a web interface allowing users to design the behaviour of an agent according to a given value of arousal and

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MuC'20 Workshops, Magdeburg, Deutschland

© Proceedings of the Mensch und Computer 2020 Workshop on «User-Embodied Interaction in Virtual Reality (UIVR)». Copyright held by the owner/author(s).
<https://doi.org/10.18420/muc2020-ws134-337>

valence, i.e. the audience's engagement and its opinion toward the speech. An important aspect of this study was to investigate the link between non-verbal behaviours and the overall perception of an audience. This first evaluation provided the desktop model we are implementing in a VR system. Moreover, they designed another web interface where users had to identify the audience's state by rating it through five-points scales, i.e. one for the valence and one for the arousal. Results afford rich details on how VA are perceived. For instance, it details how contradictory behaviours are perceived as well as how prominent behaviours are with regards to the global perception of the audience.

Finally, we took out from this model a set of variables which allows us to manipulate VA's states.

- Gaze direction and gaze away frequency;
- Postures in terms of openness and proximity;
- Head movements' type and frequency;
- Facial expressions' type and frequency;
- The number of agent displaying the target state or not.

Besides, this behaviour model is not dedicated to a context and can be enriched with specific behaviours (e.g. classroom). However, in-between agents' interactions are not taken into account within this behaviour model.

3 CONTRIBUTIONS

The contributions of this paper are as follows:

- (1) Description of the implementation of an existing atmosphere model for small and large virtual audiences in a popular open-source game engine for VR application (Section 4);
- (2) Experiment report verifying of the system's scalability, to handle up to 100 agents without degrading the VR-required frame rate (Section 5);

4 SYSTEM OVERVIEW

Our system allows to control agents' behaviour through factors initially used in the original VA behaviour model from [4]. This model has been implemented for the Unreal®Engine 4 (UE4) which is a high-end development framework widely used for 3D games and VR software. To facilitate its use, the entire system is implemented in a plugin which is compatible with most recent versions of the game engine.

Our system, named the **AtmosMaker**, is based on three main modules strongly linked to the UE4 base classes for virtual agents (Figure 1). The main module, also named *AtmosMaker* grants control of the non-verbal parameters corresponding to the model's factors, e.g. gaze, facial expression, frequency of gaze aversion. It behaves like an audience manager, enabling it to alter the audience with high level features. This *AtmosMaker* manager allows to modify parts of the VA, the entire audience or only one agent at a time.

It can be paired with a 3D graphical user interface (GUI), e.g. to edit the VA in real time directly in VR. The atmosphere can be tailored to elicit a particular emotional response for a specific application domain (e.g., virtual classroom, conference, theatre, interview). To replicate Chollet and Scherer behaviours' frequencies, all behaviours have been set relatively to a time period. For instance, the amount of time with averted gaze, e.g., 0%, 25%, 50% during a given period.

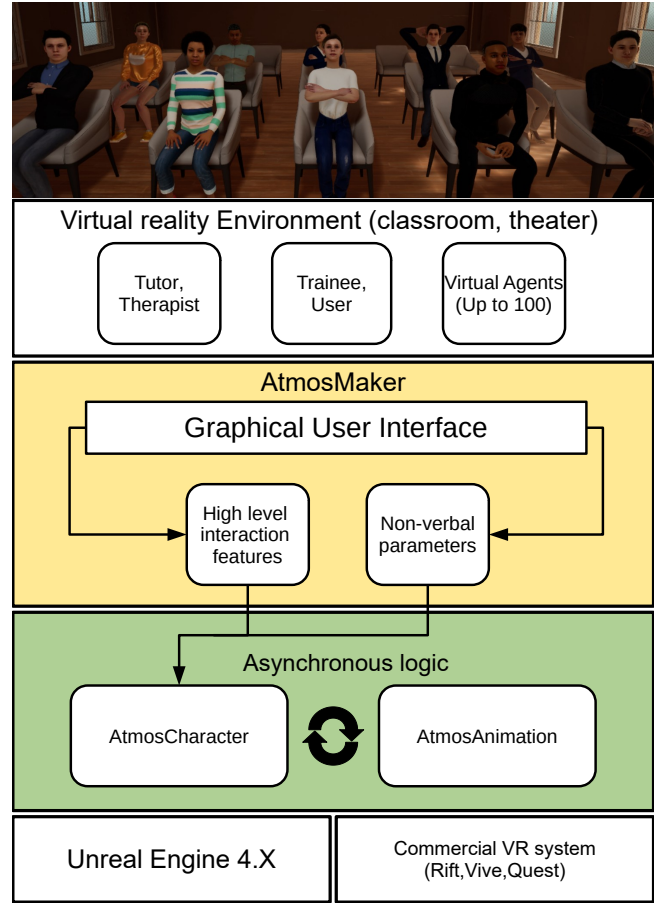


Figure 1: Current system architecture. More information and source code at <https://www.hci.uni-wuerzburg.de/projects/virtual-audiences/>.

An **AtmosAnimation** instance computes the movements of the agent according to the non-verbal parameters. The animation actually played out by an agent is dynamically constructed from one to several animations or applies rotations on character's bones, each corresponding to a specific factor from the model (e.g. the neck's bone to gaze away from the user). There is an animation instance per agent due to how UE4 animates agents. Furthermore, extending this animation instance empowers our system with advance features, e.g. blending animations, layering agent's skeleton or using inverse kinematic.

Finally, an **AtmosCharacter** class exposes the different model's factors to the *AtmosMaker* and triggers the animations blending according to their value by overseeing the *AtmosAnimation* instance. This *AtmosCharacter* inherits from the UE4's character class which allows to combine our system with all the artificial intelligence components available in the game engine, e.g. state machines, navigation, character's senses modules. Because all the behaviours available in our implementation are rarely triggered together, we are using an event system for optimisation purposes, it is described in the next subsection 4.1.1.



Figure 2: Virtual Audience composed of Mixamo's character.

If facial expressions are required, the agent's mesh must implement morph targets, i.e. blend shapes or shape keys. If it does not have any, the *AtmosAnimation* will not be able to warp the mesh. The plugin can handle model and animations from different free 3D Character Modelling tools such as Mixamo, or Autodesk Character Generator (Figure 2): it can be easily enriched by adding new animations, facial expressions and head movements.

For our early prototype we chose to use seated agents in order to fit within our research context (i.e. teachers' training). However, the system is flexible and allows using agents from different 3D Character Modelling tools, which can be placed in a variety of configurations, and play stand-up animations. The system has also been tested with different hardware and head-mounted displays (HMDs): mobile HMDs and HMDs with integrated eye-tracking are both compatible.

4.1 Operating principle

In order to animate an agent, the different states it can encounter when in animation were identified. The system is based on a loop which alternatively shows or hides the result of the animation building process.

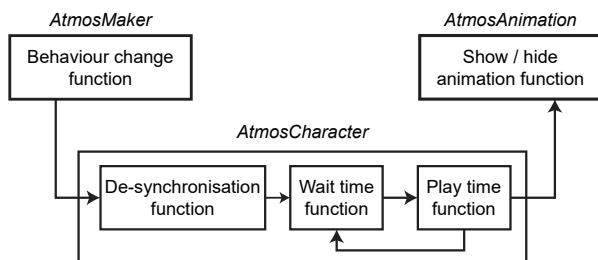


Figure 3: Animation loop's execution flow. From the *Atmos-Maker* manager to the different stages needed to follow the behaviour model and finally to the animation process.

4.1.1 Base animation loop. The animation phase has two parts (Figure 3): one waiting and another playing the animation. States are represented by independent timers duration, and the transitions between them are function calls triggered by these timers. An internal timer manager computes the time elapsed for each timers. The base loop starts with a timer that represents the period where the agent is waiting before showing any behaviour. When it reaches its end, it triggers the call to the next function in the loop and so on until all triggered behaviours were played out. Finally, the loop goes back to its wait state when this last timer completes itself.

Every timer duration is based on a fixed value called reference period. This parameter is used in every wait and play timers computation along with the corresponding display frequency factor value provided in the *AtmosMaker* module. A fixed period of time can be used to represent the rate at which we want to show a certain type of animation for an agent in the audience (e.g. gazing away from the user 60% of the period).

To avoid breaks in the virtual audience's believability, we add a timer with a random time before the animation loop's first execution. This delay is used to desynchronize the start of the agent's animation, and thus breaking a negative effect where every agent in the audience would begin to show at the same time their respective animations

As all animations are not expected to be continually executed, the system does not rely on the update function provided with the game engine. This function is internally called every frame, and constantly evaluates animations. Hence, it is severely decreasing the application's performances. Concatenating functions calling each others was the solution to avoid updating every frame.

4.1.2 Animation prioritisation. During the development of this module's component we noticed that the generated animations although diverse, did not blend well in some cases. When performed, animations could lead to physical oddities and overlapping behaviours that can break the immersion for the user, creating uncanny situations. As a solution, we decided to avoid the blending of certain animation types, e.g. the gaze direction and the head movement. But, we also wanted to keep the activation of incompatible animations types in some cases, in order to execute them when an opening becomes available (e.g. a period where the agent does not show any animation).

The solution we chose to keep is based on a simple method. The first animation which is executed by the code becomes the prioritised one, i.e. it also signifies to other animations the priority order. In case of conflict between two incompatible animations, we cancel the activation of the lower priority one. Due to asynchrony, some triggered behaviours could never be played out. To address this issue, a time threshold interrupts the prioritised animation, e.g. if an agent is gazing away for 100% of the time the threshold will stop the behaviour to let the others to run.

The model implementation we are proposing allows to control VA by triggering the different non-verbal behaviours. Our system also keeps the generated VA away from behaviours which can be considered as not believable or incongruous, e.g. synchronised animations or a head shake while a gaze direction is changing.



Figure 4: Desktop and virtual reality screenshots of the first scalability benchmark. From left to right: 2 agents with a desktop GUI example, 28 agents in a VR overview, 1000 agents with desktop GUI.

The next performance study allows to apprehend the system's limitation in terms of simulation capacity and potential improvements.

5 SYSTEM SCALABILITY PERFORMANCES

Low latency is critical for VR systems. It is a negative factor in simulator sickness, and it also considerably affects interaction [17]. Low latencies and its jitter are also critical requirements for enabling collaborative applications with VR systems using virtual agents and embodied avatars [14]. Consequently, the evaluation focuses on measuring our system's impact on the frame rate and identifying the maximum threshold number of simultaneous agents in VR we can support without any animation and mesh optimisations.

The following results are divided in two parts. The first part shows the performances with low detailed agents (Figure 4) and the second with a more detailed agents (Figure 6), approximately four times more complex in terms of polygons count.

5.1 Device

To perform the evaluation we used a laptop running with Windows 10 64 bits, Intel®Core i7-7700HQ processor (Quad core, 2.80 GHz, 8MB cache, 8 GT/s) and NVIDIA®GeForce GTX 1070 Graphics Processing Unit (GPU) 8GB GDDR5. A HTC®Vive was used to carry out the VR evaluation. We choose to carry out the evaluation with a VR ready laptop we are commonly using to run our evaluations.

5.2 Environment configuration

A simple VR scene was built without any superfluous post-processing or shadows. Thereby, it makes it easier to detect the bottleneck of our plugin. Thus, only a plane, the agents and a default sky-box was rendered (Figures 4, 6). Several settings had to be made, for instance VSync and UE4's frame smoothing were disabled because they can interfere with the frame rate in order to fit with the monitor's refresh rate. With regards to the VR head mounted display, it was at a fixed position in order to avoid rotations and thus fluctuations in the processor's draw calls. All measures are medians done on a fixed amount of time, i.e. 100 frames.

5.3 Results

Our results with the low detailed agents are visible on Figure 5 (blue curve). From the performances data we distinguish three thresholds : i) Up to 30 agents the average frame rate is high (> 90 Hz) ii) Up to 100 the frame rate decreases while remaining acceptable for VR usage ($[80, 90]$ Hz), iii) Above 200 agents, the system is no more suitable for VR use (< 30 Hz). We noticed that our system was CPU bounded. This is why we investigated which process was the

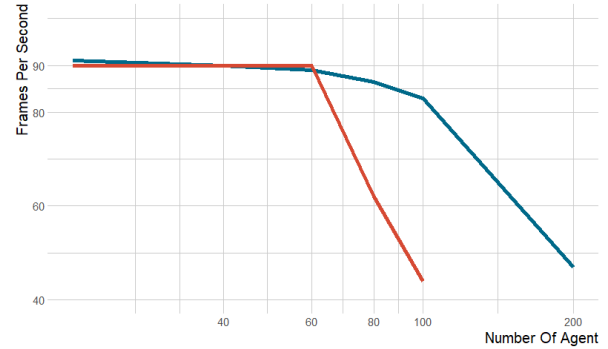


Figure 5: Frame rate depending on the number of agent. In red the complex agent scalability data. In blue the light-weight agent scalability data.

most consuming for the CPU between the draw time and the time needed to update the audience, i.e. the time to update behaviours, meshes, and animations. It appears that the render time is not the bottleneck of our system but the cumulative time to update each agents in the audience.

The second evaluation does not face the same problem. In fact, while the agents' triangles are increasing fast the number of agent does not (Figure 5 red curve). We are here facing the opposite problem with a draw time which is reducing the frame rate. With complex agents the system can handle around 60 agents before the draw time reduces the frame rate which is causing less fluidity in agents' movements for instance.

In order to make our data easier to read we are only displaying the frames per second count while the system is not yet bounded by the rendering. The step with 100 agents is also displayed in the second evaluation even if the system is already bounded by the rendering in order to compare the different impacts of the audience's update time and the draw time mainly affected by the number of characters to render. The details of the draw time and the update time are not displayed because they correspond to the frame rate given in the figure, depending on which one is the bottleneck. As a complementary fact the draw time is approximately half of the update time in our first evaluation before the draw time becomes more important, i.e. between 300 and 400 agents.

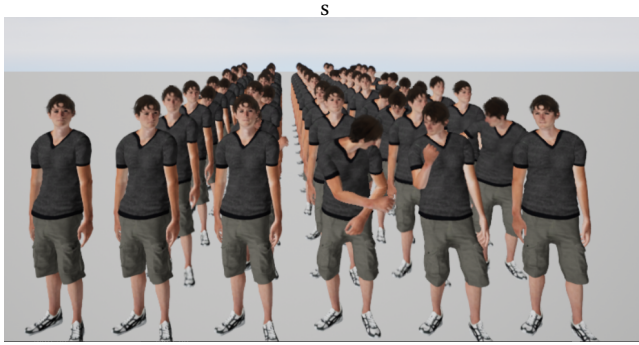


Figure 6: Benchmark environment view. Close view of 60 Mixamo's Remy agent.

6 DISCUSSION

The *AtmosMaker* offers a high-level building tool for audiences' atmosphere. The implementation described in section 4 may offer to the VR community the opportunity to design and control VA with precise parameters from the literature. Considering the scalability data, the system seems to indicate suitable performances for VR if the system stays under the threshold of 100 agents and under 60 with more complex agents (Figure 5).

The results described are closely linked to the chosen hardware. A more performing one would allow improving our system's performances. However, as explained in the previous section this benchmark allows us to better apprehend the audiences' size the *AtmosMaker* can handle with a standard material commonly use in our user evaluations. Moreover, in the case where larger audiences would be needed, different solutions frequently used in VR games can be used. For instance the use of level of details (LOD) to simplify meshes in terms of polygons count. Reducing the number of agents' bones and only updating agents' pose when rendered are alternatives. These improvements could significantly enhance performances and more precisely CPU load when rendering the virtual environment. Regarding the audience's update time which is mainly affected by the layered animations, only updating the pose when the agents are rendered could help as well as improving the animation algorithm, i.e. mixing animations asks more resources.

VR GUIs as shown in Figure 7 seem to offer the possibility to easily design atmospheres. These interfaces also authorise in-game modifications, which could be used by systems using audiences like [3, 9, 16].

If other domains may gain advantage from a VA behaviour system, this current implementation is still under evaluation, e.g. performances, audience perception and agents' believability. The *AtmosMaker* could also benefit VR training systems, with regards to the trade-off between a fully autonomous simulation and a completely Wizard of Oz system where each agent would have to be individually controlled: for applications where replacing tutor expertise with an autonomous component is not desirable, our system can provide a component with high-level controls which may be easily used by the tutor during a simulation-based training session.

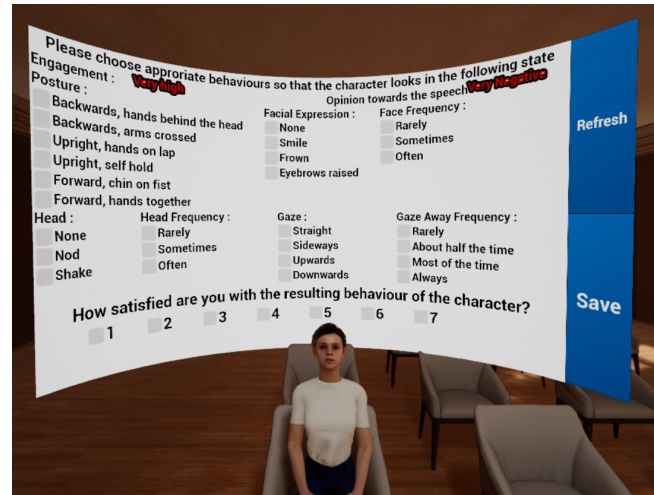


Figure 7: Example of a graphical user interface in virtual reality used in a user evaluation aiming at reproducing Chollet and Scherer's results.

7 FUTURE WORK

With knowledge of *AtmosMaker*'s limitations, we are now able to run user perception evaluations in order to replicate desktop results [4]. Assessing the system capability to generate atmospheres and investigating if users perceive them as intended is a compulsory step in our long term study. Figure 7 is an example of Chollet and Scherer's evaluation replication in VR where users have to design agent's behaviour according to the valence and arousal dimensions.

In terms of interactivity and awareness the system is still limited, it requires a user to trigger interactions or changes in the VA, i.e. similar to Wizard of Oz systems. Therefore, we are planning to use interactive storytelling techniques to obtain more realistic and variable VR training systems. They should bring control over the scenario, e.g. with narrative events and VA's behaviour planning. Within the context of teacher training systems the model we are using [4] could be enhanced with context-related animations and backchannels in order to bring more believability and responsiveness to the VA. A practical example of backchannels are sounds that Chollet and Scherer's model does not consider. In fact, in terms of immersion and believability the sound should not be neglected; background noises and phatic expressions like 'yeah' or 'mmhmm' would bring more depth to the simulation. These improvements could help instructors to manage more precisely and to make the classroom's level of attention and approval more noticeable for the user.

These backchannels can be triggered by eye-tracking data, by end of sentence detection, by voice analysis, from the users locations or even from physiological sensors for stress detection (e.g. skin conductance, temperature, heart rate, pupils dilatation). It should allow instructors to follow their training scenario without being overwhelmed by the supervision of the simulation consistency, i.e. by letting the system handling these backchannels automatically.

Such model improvements will need to be evaluated within a training system context in order to testify to their effectiveness. The

interactive narrative engine we are planning to develop will also need evaluation in terms of scalability performances and instructors acceptability.

8 CONCLUSION

In this study we presented a VA Atmosphere model implementation and its limitations. As far as the scalability data goes, the system allows to control approximately 100 agents in VR or 60 high quality agents without compromising the frame rate. The *AtmosMaker* offers a high-level building tool for audiences' atmosphere. This tool gives the VR community the opportunity to design audiences within a game engine. We believe having a high-level VA programming tool will benefit training or therapeutic applications by bringing the capability to manipulate and create the rendered atmosphere of an audience quickly and easily. Eventually, our next evaluation aims at testifying to the quality of the generated atmospheres in VR. This should lead to our next implementation of a teacher training system using storytelling engine.

REFERENCES

- [1] Kevin W Arthur, Kellogg S Booth, and Colin Ware. 1993. Evaluating 3d task performance for fish tank virtual worlds. *ACM Transactions on Information Systems* 11, 3 (1993), 239–265.
- [2] Ligia Batrinca, Giota Stratou, Ari Shapiro, Louis-Philippe Morency, and Stefan Scherer. 2013. Cicero - Towards a Multimodal Virtual Audience Platform for Public Speaking Training. In *Intelligent Virtual Agents*, Ruth Aylett, Brigitte Krenn, Catherine Pelachaud, and Hiroshi Shimodaira (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 116–128.
- [3] Mathieu Chollet, Pranav Ghate, and Stefan Scherer. 2018. A Generic Platform for Training Social Skills with Adaptive Virtual Agents. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (Stockholm, Sweden) (AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1800–1802. <http://dl.acm.org/citation.cfm?id=3237383.3237982>
- [4] Mathieu Chollet and Stefan Scherer. 2017. Perception of Virtual Audiences. *IEEE computer graphics and applications* 37, 4 (2017), 50–59.
- [5] Mathieu Chollet, Giota Sratou, Ari Shapiro, Louis-Philippe Morency, and Stefan Scherer. 2014. An interactive virtual audience platform for public speaking training. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1657–1658.
- [6] Masato Fukuda, Hung-Hsuan Huang, and Toyoaki Nishida. 2018. Investigation of Class Atmosphere Cognition in a VR Classroom. In *Proceedings of the 6th International Conference on Human-Agent Interaction (Southampton, United Kingdom) (HAI '18)*. ACM, New York, NY, USA, 374–376. <https://doi.org/10.1145/3284432.3287191>
- [7] Masato Fukuda, Hung-Hsuan Huang, Naoki Ohta, and Kazuhiro Kuwabara. 2017. Proposal of a Parameterized Atmosphere Generation Model in a Virtual Classroom. In *Proceedings of the 5th International Conference on Human Agent Interaction*. ACM, 11–16.
- [8] Kelly S Hale and Kay M Stanney. 2006. Effects of low stereo acuity on performance, presence and sickness within a virtual environment. *Applied Ergonomics* 37, 3 (2006), 329–339.
- [9] Sandra R Harris, Robert L Kemmerling, and Max M North. 2002. Brief virtual reality therapy for public speaking anxiety. *Cyberpsychology & Behavior* 5, 6 (2002), 543–550.
- [10] Aleshia T. Hayes, Stacey E. Hardin, and Charles E. Hughes. 2013. *Virtual, Augmented and Mixed Reality. Systems and Applications: 5th International Conference, VAMR 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter Perceived Presence's Role on Learning Outcomes in a Mixed Reality Classroom of Simulated Students, 142–151. https://doi.org/10.1007/978-3-642-39420-1_16
- [11] Claudia Hendrix and Woodrow Barfield. 1996. Presence within virtual environments as a function of visual display parameters. *Presence: Teleoperators & Virtual Environments* 5, 3 (1996), 274–289.
- [12] Ni Kang, Willem-Paul Brinkman, M Birna van Riemsdijk, and Mark Neerincx. 2016. The design of virtual audiences: noticeable and recognizable behavioral styles. *Computers in Human Behavior* 55 (2016), 680–694.
- [13] Marc Erich Latoschik, Florian Kern, Jan-Philipp Stauffert, Andrea Bartl, Mario Botsch, and Jean-Luc Lugin. 2019. Not Alone Here?! Scalability and User Experience of Embodied Ambient Crowds in Distributed Social Virtual Reality. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2134–2144.
- [14] Marc Erich Latoschik, Jean-Luc Lugin, Michael Habel, Daniel Roth, Christian Seufert, and Silke Grafe. 2016. Breaking Bad Behavior: Immersive Training of Class Room Management. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology (VRST '16)*. ACM, New York, NY, USA, 317–318.
- [15] Cha Lee, Scott Bonebrake, Doug A Bowman, and Tobias Höllerer. 2010. The role of latency in the validity of AR simulation. In *2010 IEEE Virtual Reality Conference (VR)*. IEEE, 11–18.
- [16] Jean-Luc Lugin, Marc Erich Latoschik, Michael Habel, Daniel Roth, Christian Seufert, and Silke Grafe. 2016. Breaking Bad Behaviours: A New Tool for Learning Classroom Management using Virtual Reality. *Frontiers in ICT* 3 (2016), 26.
- [17] Jean-Luc Lugin, Dennis Wiebusch, Marc Erich Latoschik, and Alexander Strehler. 2013. Usability benchmarks for motion tracking systems. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology (VRST '13)*. ACM, 49–58.
- [18] Michael Narayan, Leo Waugh, Xiaoyu Zhang, Pradyut Bafna, and Doug Bowman. 2005. Quantifying the benefits of immersion for collaboration in virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 78–81.
- [19] David-Paul Pertaub, Mel Slater, and Chris Barker. 2002. An experiment on public speaking anxiety in response to three different types of virtual audience. *Presence: Teleoperators & Virtual Environments* 11, 1 (2002), 68–78.
- [20] Elisa Shernoff, Stacy Frazier, Christine Lisetti, Cedric Buche, Stephanie Lunn, Claire Brown, Alban Delmarre, Tommy Chou, Joseph Gabbard, and Emily Morgan. 2018. Early Career Teacher Professional Development: Bridging Simulation Technology with Evidence-Based Behavior Management. *Journal of Technology and Teacher Education* 26, 2 (April 2018), 299–326.
- [21] Anthony Steed. 2008. A Simple Method for Estimating the Latency of Interactive, Real-time Graphics Simulations. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST '08)*. ACM, New York, NY, USA, 123–129.