Multi-agent Crowd Simulation in an Active Shooter Environment

Sharad Sharma(✉) and Syed Ali

Department of Computer Science, Bowie State University, Bowie, MD, USA
ssharma@bowiestate.edu, alis0712@students.bowiestate.edu

AQ1

Abstract. In recent years there has been a sharp increase in active shooter events, but there has been no introduction of new technology or tactics capable of increasing preparedness and training for active shooter events. This has raised a major concern about the lack of tools that would allow robust predictions of realistic human movements and the lack of understanding about the interaction in designated simulation environments. It is impractical to carry out live experiments where thousands of people are evacuated from buildings designed for every possible emergency condition. There has been progress in understanding human movement, human motion synthesis, crowd dynamics, indoor environments, and their relationships with active shooter events, but challenges remain. This paper presents a virtual reality (VR) experimental setup for conducting virtual evacuation drills in response to extreme events and demonstrates the behavior of agents during an active shooter environment. The behavior of agents is implemented using behavior trees in the Unity gaming engine. The VR experimental setup can simulate human behavior during an active shooter event in a campus setting. A presence questionnaire (PQ) was used in the user study to evaluate the effectiveness and engagement of our active shooter environment. The results show that majority of users agreed that the sense of presence was increased when using the emergency response training environment for a building evacuation environment.

Keywords: Virtual reality · Active shooter events · Crowd simulation · Multi-agent simulation

1 Introduction

Crowd simulation has many uses, which include improving traffic flows in busy highways and streets [1], enhancing training and virtual environments, and implementing artificially intelligent (AI) characters in games and movies [2]. Crowd behavior during an active shooter simulation has been an important topic of interest as it aims to reduce casualties during emergency event. In the United States and around the world, reported active shooter incidents have become incredibly common. In the year 2020, there have been over 40 active shooter incidents in the United States, with over 200 fatalities [3]. Due to this, it is critical to find tools and technologies that will allow for the modeling and simulation of human behavior during emergency response training. The law enforcement personnel must be properly trained for an active shooter event so that they

J. Y. C. Chen and G. Fragomeni (Eds.): VAMR 2022, LNCS 13318, pp. 1–13, 2022.
https://doi.org/10.1007/978-3-031-06015-1_8

can get a better overview of the situation and perform the best response strategy. VR training environment allows one to perform virtual evacuation drills for different what-if situations without any accompanying risk. Traditional performance-based tests for emergency evacuation drills for fire and active shooter response are expensive to perform due to safety and legal issues.

Fig. 1. Active shooter response environment for a campus building

According to the FBI [4], most of the active shooter incidents have taken place at educational facilities such as high schools or university campuses between the years 2001 and 2015. The 2007 Virginia Tech massacre is a prime example of an active shooter event at a higher educational facility, where 33 individuals lost their lives, along with 127 injured, which is why in this study we propose a system that simulates agent behavior when an active shooter is present at a university campus. This paper presents a VR training module for active shooter events for a campus building emergency response in an institute of higher education (IHE). The VR environment is implemented in Unity 3D where the user has an option to enter the environment as a security personnel or as an occupant in the building. The VR training module offers a unique way of performing virtual evacuation drills for different what-if scenarios. The novelty of our work lies in the implementation of crowd behavior in the VR active shooter environment. We have presented two ways for controlling crowd behavior. First, by defining rules for agents or NPCs (Non-Player Characters). Second, by providing controls to the users as avatars or PCs (Player characters) to navigate around the virtual environment as autonomous agents using a keyboard when an active shooter is present. The rules are developed by implementing behavior trees for the agents. The system is built using the Unity gaming engine. Figure 1 shows our developed VR training environment for active shooter events for the course of action, visualization, and situational awareness.

The rest of the paper is organized as follows: Sect. 2 discusses the existing work that has been done related to crowd simulations, Sect. 3 discusses how the system was implemented in Unity and the hardware specifications used to build the system, Sect. 4

discusses the user studies and evaluation of the system using presence questionnaire (PQ) scheme and lastly, Sect. 5 discusses the drawn conclusions and proposed future work. Finally, Sect. 6 states acknowledgments and references.

2 Related Work

Lately, there has been an increase in active shooter events. As a result, there is a need to learn from past active shooter events and create public awareness for the safety procedures and tactics. Hoogendoorn et al. [5] have demonstrated a behavior system of three interrelated layers:

- Operational layer: It describes the movement of agents such as walking, acceleration, and approaching a point of interest.
- Tactical layer: This is used in pathfinding algorithms, collision evasion, and route preparation.
- Strategic layer: This shows how the agents will choose their next point of interest based on the existing state of the model environment.

Currently, many frameworks can be used to simulate crowd behavior. Most of these frameworks typically share the same set of common operational and tactical models of crowd behavior [6] which includes ORCA (Optimal Reciprocal Collision Avoidance) [7] or GCF (Generalized Centrifugal Force) [8, 9]. These frameworks provide a reliable crowd simulation behavior. The issue with these frameworks is that the behavior patterns are the same for all agents. For example, the agents will always choose the shortest route to their targets, and the behavior only varies when there is an obstacle on the way. Singh et al. [10] have created an agent-centered system that was used for simulating evolving behaviors. It was designed to detect and classify emerging agent behavior to control undesirable actions. Agents in this framework followed an OODA (Observe, Orient, Decide, Act) loop, and their behavior system was based upon an FSM (Finite-State Machine). FSM is a good tool for developing looped behaviors, but the system is restricted and cannot create believable characters.

Agent-centered pedestrian simulation system allows users to implement, test, and develop various behavior models. MomenTUM [11, 12] is a Java-based framework and can expand the set of used models and allows users to add different models in a single simulation. The system also allows users to add different behaviors, such as an awareness system, to their agents. The framework comes with a graphical user interface (GUI) built-in that allows users to configure their simulation environment without coding. Thus, MomenTUM allows users to create a flexible, and generalized approach to crowd simulation and provides a generalized view on agent's movements. The issue with this framework is that it doesn't create an immersive environment to fully simulate everyday life. The agents in this framework cannot interact with everyday objects in the model environment, and there is no support for animation states, which is crucial when creating believable characters.

There has been considerable interest in modeling and simulation of human behavior for emergency response during evacuation [13]. Sharma et al. [14, 15] have created an active shooter response training environment for a building evacuation in a collaborative virtual environment (CVE). Their CVE is implemented in Unity 3D and is based on run, hide, and fight mode for emergency response. The participant can enter the CVE setup on the cloud and participate in the active shooter response training environment. CVE has been used as a training and education tool for many applications such as military, psychological, medical, and education applications, subway evacuation [16–21]. AnyLogic [22] is a crowd simulation modeling framework that supports discrete-event, system dynamics pedestrian simulation, and agent-centered pedestrian simulation systems. This system is widely used to simulate commercial applications such as markets, healthcare, and manufacturing. The system also contains a pedestrian library which is used to envision pedestrian flows in models. The library also can gather statistics on pedestrian traffic (crowd density) in various areas of the model. AnyLogic utilizes flowcharts to set the behavior of the agents. AnyLogic is has a limited set of possible actions, so the agent can only move from one point to another. It is therefore a powerful tool to get statistics about pedestrian activity and was not designed for an immersive pedestrian system [23].

Menge [24] is a widely used open-source crowd movement C++ framework designed for pedestrian dynamics. It allows users to decompose the agent behavior into three sub-categories which include: goal selection, planned computations, and planned variations. Menge utilizes a set of pathfinding algorithms and movement parameters that generates an accurate pedestrian behavior system. Goal selection in Menge is implemented using Finite State Machine (FSM). Modern games use behavior trees [25] for implementing AI to non-playable characters. Most gaming engines such as Unity come with a built-in behavior tree module. This module has a hierarchical structure, where each node on the agent has a specific behavior implemented to it and is executed once it is initialized. This allows users the ability to create complex behaviors on each or group of agents.

3 Crowd Simulation Behavior Framework

Our proposed active shooter VR simulation is created using the Unity gaming engine. Unity provides the ability to achieve complex agent behaviors with high-quality resolution. Our proposed crowd simulation behavior framework includes the following functional requirements:

1. The agents within the system should have the ability to make decisions.
2. The system should be able to simulate crowds, adjust their speeds accordingly and create goal posts or points of interest for the agent to run to when an active shooter is around.
3. System should have the capability of implementing animated non-playable agents.
4. Behavior AI model on one agent should easily be expandable to be used in other models.

Behavior trees have a wide range of applications. A behavior tree is a tree of ranked nodes on an AI object that control the flow of decision making (strategic layers) and forms branches on various types of weighted (utility) nodes that control the agent's intentions. As we progress down the tree, it can reach the sequences of instructions that would be best suited to the situation [26]. When game developers are creating AI for games, they often face situations like what AI path the agent should take next from a given set of possible paths. If developers, for instance, decided to use a utility-based algorithm, then each action must be matched with a utility curve [27]. In our crowd simulation behavior system, agents normally choose a path based on their intention of their inner state. To achieve this, several goal points were added to the model world. To choose the best goal point, behavior trees are implemented to the agent, where the weighted nodes then decide the best path to take. Figure 2 shows how the behavior trees were implemented in our system.

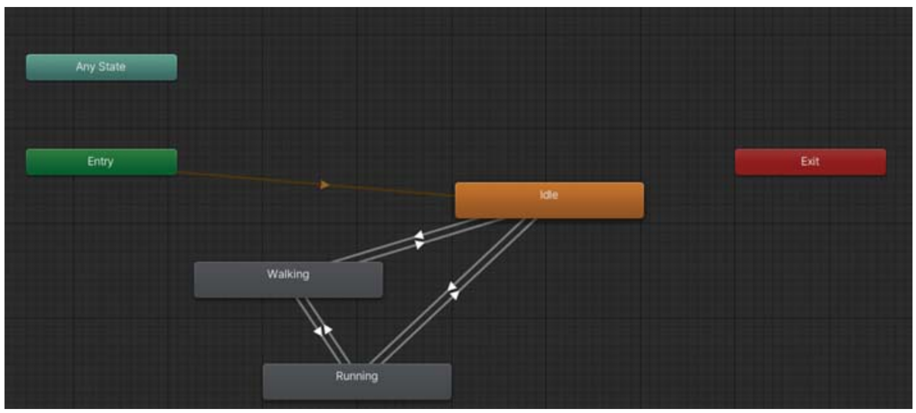

Fig. 2. Implementation of behavior trees in our crowd simulation system. Each node is weighted so that agents can choose the best goal point.

Decision-making is a key component of behavior trees. It allows users to describe how the agent will choose the next state or action. The choice is defined by the agent's utility nodes. Intention based approach [28] has been widely used for defining utility nodes. Intention describes the possible actions taken by the agent. To determine the utility intention node, an "urge" value (weights) was placed for each node in our developed decision-making system. This value determines which intention needs to be done first and is changed over time based on other factors. Intention can be divided into two subcategories:

1) Mandatory: These intentions must be followed by every agent in the model during the simulation and is usually at the parent node of the behavior tree. An example is if there is an active shooter within the vicinity then all agents surrounding the vicinity will run away from the shooter.
2) Standard: These are optional intentions and may or may not be followed by every agent during the simulation. For example, when an active shooter is not present in the vicinity then some agents might be walking, or sitting, or running to catch a bus, and so forth.

Other types of intention include Uninterrupted Intention: Here the agent's behavior will not change. For example, even with an active shooter around the vicinity, some agents will not be running away from the shooter since they will be busy performing a specific task. In our system, the agents are separated into different behavior groups for different intentions. However, common characteristics such as weight, speed, and radius are still shared between all the agents. The behavior group determines how the objects in the model world would behave with the agent. Agents or groups of agents may have different interaction logic implemented when interacting with certain objects. We can consider the whole decision-making process as the brain of each agent since it accumulates and maintains all the agent's intentions. Figure 3 demonstrates our decision-making system:

Fig. 3. 3D Data visualization in non-immersive environment

To support the agent's interaction in the model world, certain points of interest or goal posts were added. This allows agents to randomly move around the points of interest or goal posts. Once the agent reaches a certain goal post or interest point, the process of interaction starts. During the interaction process, the agent's behavior tree nodes start to randomly move around. Once the interaction stops the structure of the behavior tree changes, and a new intention state begins.

Fig. 4. An example of an interaction space. Different agents have different behaviors within a small space.

As described by T. Plch, et al. [25], two types of points of interest are added to the model which are interaction objects and interaction areas. Interaction objects are objects that are placed into the model world and allow agents to interact with the object using intention (animation) states. In our system, a specified number of agents can only interact with one object, and their interaction time is also limited. Interaction areas are an expansion of interaction objects where agents occupy a small space within the model and interact with that space a certain way. Figure 4 shows an example of an interaction space where agents have different behaviors. Thus, our system's whole decision-making process of an agent can be described as follows:

- Creating a behaviour tree with all possible intentions.
- Placing weights for each intention.
- Choosing an intension with the highest weight.
- Picking the appropriate goal post.
- Moving to that goal post.

4 Crowd Behavior Implementation

Our proposed VR environment is designed to simulate crowd behavior during an active shooter event at the university campus. We have built a scenario to simulate an active shooter response inside the building. The environment is built using the Unity game engine and used the behavior tree module within the engine for building the character's AI framework. Before to the active shooter arriving on the scene, the agents move around to random points of interest.

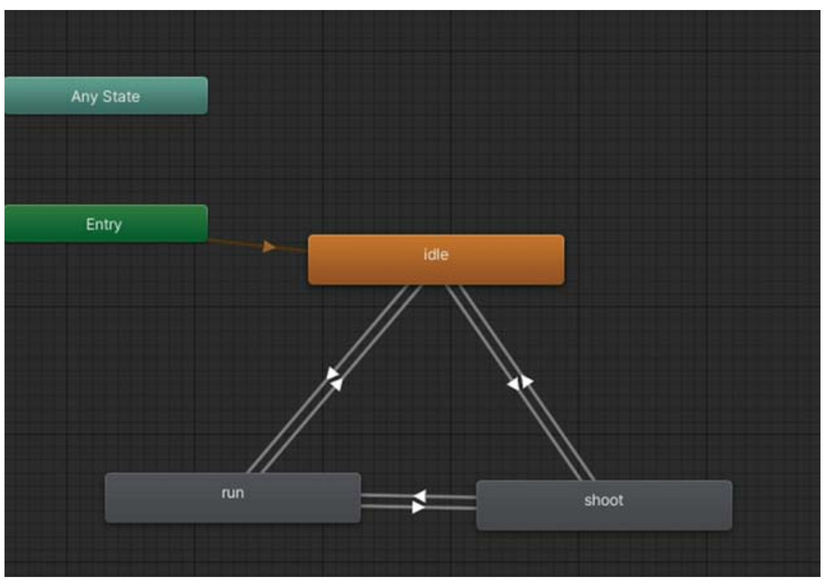

Fig. 5. Interacting with the map through Oculus touch controllers.

We have implemented several goal points or points of interest in Unity, and the agents tend to pick random goal points while moving around the environment. While they are moving around random points of interest, the intention is set to a "walk and talk" state. We have also implemented an audio functionality for the agents so that they can talk while moving around to different goal states in the environment. The audio functionality adds an extra layer of immersion to the simulated environment. Once the user presses the "Q" key on the keyboard it spawns in the active shooter. The intention state for the active shooter is set to "idle", however, the intention state for the crowd changes from "walk and talk" state to "run and talk" state, and we notice that the crowd moves away from the active shooter. We also notice that after a certain distance, the crowd's intention changes back from "run and talk" state to "walk and talk state". When the active shooter spawns in the agents pick random points of interest away from the active shooter. An example of the active shooter's intention state is shown in Fig. 5. However, the intention state for the agent changes for example "talk" state, to "hide" state.

Table 1. Framework characteristics summary

CHARACTERISTICS	FRAMEWORK
ENGINE	Unity
PROGRAMMING LANGUAGE	C#
DECISION-MAKING SYSTEM	✓
DYNAMIC AGENTS	✓
INTENSION SUPPORT	✓

As mentioned previously the decision-making tree was implemented in Unity. Unity comes with a built-in C# and behavior tree framework which allows us to trigger certain intentions and control the behavior of the agents. Table 1 summarizes the characteristics of our framework.

5 User Studies and Evaluation

This paper presents crowd behavior during an active shooter event. We have presented how the agents behave during an active shooter in the model world and compare it with a real-world scenario. A presence questionnaire (PQ) was used based on Witmer et al. and Singer et al. [29] presence questionnaire in the user study to evaluate the effectiveness and engagement of our immersive active shooter environment. A limited user study was conducted for this evaluation. A total of 7 participants were chosen for this study which included 5 males and 2 females. Participants were asked to fill out a short survey at the end of the study. The participants were asked a series of 11 questions. The questions were based on a presence questionnaire (PQ) framework and are outlined below. Each question on the questionnaire was answered using a 7-item Likert scale. Tables 2 outline the questions asked, the most common score, and the average response score for each question:

Table 2. Presence Questionnaire (PQ) framework

QUESTION NUMBER	QUESTIONS	MOST COMMON SCORE	AVERAGE
1	How responsive were the agents when an active shooter was present?	6,7	6.4
2	How natural was the flow of agents prior to the shooter arriving on the scene?	5,6,7	6.0
3	How natural was the flow of agents after the shooter after the shooter arrived on the scene?	3,4,5	3.9
4	How much did the visual aspects of the simulation involve you?	5	5.0
5	How much did the auditory aspects of the crowd simulation environment involve you?	3,4,6	3.6
6	How well could you identify different audios?	2,4,5	3.7
7	Were you able to localize sounds in the environment?	1,3,7	3.3
8	Did you experience any delay when the "Q" key was pressed?	6,7	6.6
9	Were you able to anticipate what would happen prior to pressing the "Q" key on the keyboard?	4,5,7	5.6
10	How quickly did you adjust to the virtual environment experience?	5,7	6.0
11	How much did your experiences in the virtual environment seem consistent with your real-world experiences?	1,2,3,4	2.6

Figures 6 show the bar graph visualizing how the scores were spread between each of the questions asked in the survey. Based on the bar graph and Table 2 we can see that the mean score value for questions 1, 2, 4, 8, 9, and 10 is approximately 5.8, showing that most participants agreed that the agents in the system were responsive. The flow of agents was natural and was able to predict what the agent's response would be when an active shooter is present. The system experienced no delays from keyboard inputs. As we move onto questions 3, 5, 6, 7, and 11 we notice that the mean score value is around 3.6 showing that most participants didn't find the flow of agents natural when an active shooter was present. Some users couldn't identify and got immersed in the environment with different sound effects. Most users weren't able to localize the sounds in the environment. Lastly, the users didn't find the virtual environment to be coherent with real-world situations. When we implemented sound effects on the agents in Unity, we used a logarithmic scale as opposed to a linear scale, which is why most participants were not able to hear localized sounds in the environment. One of the reasons why we chose a logarithmic scale was because we wanted to make the environment feel more natural, as the sounds would dissipate over distance by using a logarithmic scale. With a linear scale, however, the sound effects are discrete and so they feel less natural. However, they are good at localizing sounds. When we implemented behavior trees into our system, we only added limited behaviors which included "run", "walk" and "idle" states when an active shooter was present. This is because the machine on which the AI behavior system was built wasn't powerful enough to create intelligent behavioral AI systems. AQ2

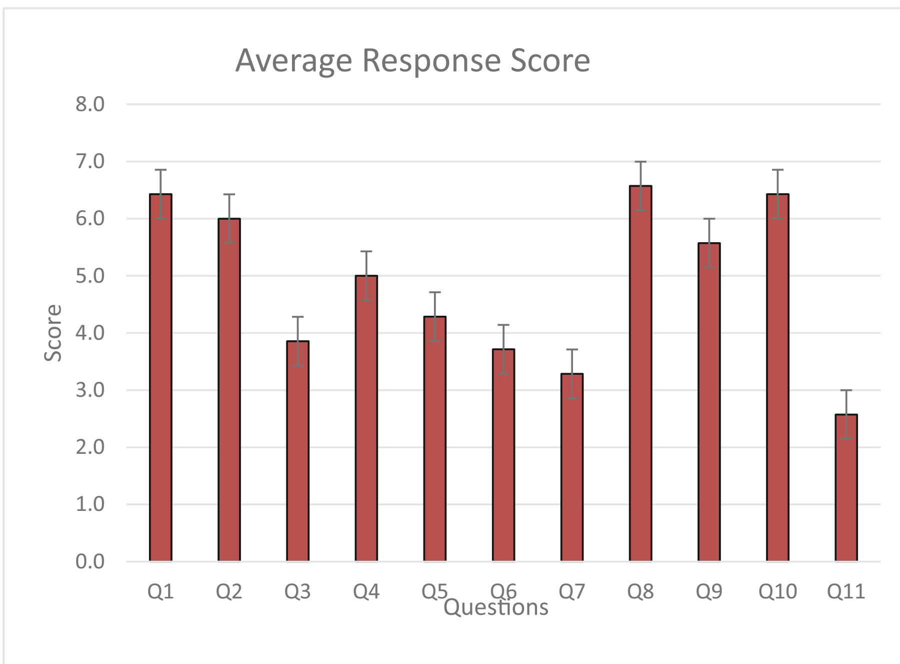

Fig. 6. Bar graphs displaying the average response scores

6 Conclusions

In conclusion, we have presented a virtual reality (VR) experimental setup where experiments for active shooter response can be conducted using computer-controlled (AI) agents and user-controlled agents. The setup can simulate human behavior during an active shooter event in a campus setting. The techniques presented in this paper include the ability of agents to move from one goal post to another (pathfinding), a decision-making system, and intention states, providing an approach to creating a simple crowd behavior framework when an active shooter is present. In contrast to other frameworks, the decision-making model implemented in our system meets all the modern requirements for developers to create their custom behaviors (depending on the PC specifications) since we use behaviors trees to implement our AI in the agents. The system is developed with the intent of having flexibility and scalability of all the components implemented into the system. The behavior tree is modeled in a way that new behavior structures can be implemented easily for different agents. This VR experimental setup can be used as a teaching and educational tool for navigation and performing VR evacuation drills for active shooter response.

Future work will involve achieving more realistic behaviors for the active shooter by adding deep learning methods to the decision-making tree. It will require behaviors to be implemented using utility AI since the algorithm will be able to adjust their utility curves for each agent to create different reactions for each intention state and will vary the behaviors according to the global intention state. Our proposed crowd simulation system is not just limited to active shooter scenarios but can be used for other emergency response scenarios such as bomb blasts or fire drills.

Acknowledgments. This work is funded by the NSF award 2131116, NSF Award: 2026412, and in part by NSF award 1923986.

References

1. Tang, M., Hu, Y.: Pedestrian simulation in transit stations using agent-based analysis. Urban Rail Transit. **3**(1), 54–60 (2017). https://doi.org/10.1007/s40864-017-0053-5
2. Chen, W.-K.: In Linear Networks and Systems, Brooks/Cole Engineering Division, pp. 123–135 (1983)
3. Active shooter simulations: an agent-based model of civilian response strategy, Dissertation, Iowa State University Digital Repository (2017)
4. Federal Bureau of Investigation. Active Shooter Incidents in the United States in 2020 | Federal Bureau of Investigation (2021). https://www.fbi.gov/file-repository/active-shooter-incidents-in-the-us-2020-070121.pdf/view. Accessed 9 Nov 2021
5. Hoogendoorn, S., van Wageningen-Kessels, F., Daamen, W., Duives, D., Sarvi, M.: Continuum theory for pedestrian traffic flow: local route choice modelling and its implications. Transp. Res. Procedia **7**, 381–397 (2015)
6. Chen, X., Li, H., Miao, J., Jiang, S., Jiang, X.: A multi-agent-based model for pedestrian simulation in sub-way stations. Simul. Model. Pract. Theory **71**, 134–148 (2017)
7. Manley, M., Kim, Y., Christensen, K., Chen, A.: Airport emergency evacuation planning: an agent-based simulation study of dirty bomb scenarios. IEEE Trans. Syst. Man Cybern. Syst. **46**(10), 1390–1403 (2016)
8. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282–4286 (1995)
9. Park, J., Rojas, F., Yang, H.: A collision avoidance behavior model for crowd simulation based on psycho-logical findings. Comput. Anim. Virtual Worlds **24**(3–4), 173–183 (2013)

10. Singh, S., Lu, S., Kokar, M.M., Kogut, P.A.: Detection and classification of emergent behaviors using multi-agent simulation framework (WIP). Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems (MSCIAAS 2017) (2017)
11. MomenTUM v2 pedestrian simulator DE - Lehrstuhl für. https://www.cms.bgu.tum.de/de/17-research-projects/99-momentum-v2-pedestrian-simulator-de. Accessed 08 Nov 2021
12. Kielar, P.M., Biedermann, D.H., Borrmann, A., Kielar, P.M., Biedermann, D.H., Borrmann, A.: MomenTUMv2: A Modular, Extensible, and Generic Agent-Based Pedestrian Behavior Simulation Framework, MomenTUMv2, pp.1–34 (2016)
13. Sharma, S., Avatarsim: a multi-agent system for emergency evacuation simulation. J. Comput. Meth. Sci. Eng. **9**(1, 2), pp. S13–S22, ISSN 1472–7978 (2009)
14. Sharma, S., Bodempudi, S.T., Scribner, D., Grazaitis, P.: Active Shooter response training environment for a building evacuation in a collaborative virtual environment. In: IS&T International Symposium on Electronic Imaging (EI 2020), in the Engineering Reality of Virtual Reality. Burlingame, California (2020). https://doi.org/10.2352/ISSN.2470-1173.2020.13.ERVR-223
15. Sharma, S., Bodempudi, S.T.: Immersive virtual reality training module for active shooter events. In: Proceedings of the IS&T International Symposium on Electronic Imaging (EI 2022), in the Engineering Reality of Virtual Reality. Burlingame, California (2022)
16. Sharma, S., Bodempudi, S.T., Reehl, A.: Virtual Reality Instructional (VRI) module for training and patient safety. In: IS&T International Symposium on Electronic Imaging (EI 2021), in the Engineering Reality of Virtual Reality, pp. 178-1–178-6(6) (2021). https://doi.org/10.2352/ISSN.2470-1173.2021.13.ERVR-178
17. Sharma, S., Bodempudi, S.T.: Situational Awareness of COVID Pandemic data using Virtual Reality, IS&T International Symposium on Electronic Imaging (EI 2021), in the Engineering Reality of Virtual Reality, pp. 177-1–177-6(6). Burlingame, California (2021). https://doi.org/10.2352/ISSN.2470-1173.2021.13.ERVR-177
18. Sharma, S., Otunba, S.: Virtual reality as a theme-based game tool for homeland security applications. In: Proceedings of ACM Military Modeling and Simulation Symposium (MMS11), pp. 61–65. Boston, MA, USA (2011)
19. Sharma, S., Otunba, S.: Collaborative virtual environment to study aircraft evacuation for training and education. In: Proceedings of IEEE, International Workshop on Collaboration in Virtual Environments (CoVE -2012), as part of The International Conference on Collaboration Technologies and Systems (CTS 2012), pp. 569–574. Denver, Colorado, USA (2012)
20. Sharma, S., Vadali, H.: Simulation and modeling of a virtual library for navigation and evacuation. In: MSV 2008 - The International Conference on Modeling, Simulation and Visualization Methods, Monte Carlo Resort, Las Vegas, Nevada, USA (2008)
21. Sharma, S., Jerripothula, S., Mackey, S., Soumare, O.: Immersive virtual reality environment of a subway evacuation on a cloud for disaster preparedness and response training. In: Proceedings of IEEE Symposium Series on Computational Intelligence (IEEE SSCI), pp. 1–6. Orlando, Florida, USA (2014). https://doi.org/10.1109/CIHLI.2014.7013380
22. Anylogic.com. AnyLogic: Simulation Modeling Soft-ware Tools & Solutions for Business (2021). https://www.anylogic.com/. Accessed 2 Jan 2022
23. Kielar, P.M., Borrmann, A.: Spice: a cognitive agent framework for computational crowd simulations in complex environments. Autonom. Agent. Multi-Agent Syst. **32**(3), 387–416 (2018). https://doi.org/10.1007/s10458-018-9383-2
24. Curtis, S., Best, A., Manocha, D.: Menge: a modular framework for simulating crowd movement, Collective Dynamics, vol. 1 (2016)
25. Li, X., Gui, X.: Modelling autonomic and dynamic trust decision-making mechanism for large-scale open environments. Int. J. Comput. Appl. Technol. **36**(3/4), 297 (2009)
26. Kallmann, M., Thalmann, D.: Modeling behaviors of interactive objects for real-time virtual environments. J. Vis. Lang. Comput. **13**(2), 177–195 (2002)

27. Bryson, J.: Intelligence by design: Principles of modularity and coordination for engineering complex adaptive agents, Ph.D. dissertation. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA (2001)
28. Cerny, M., Plch, T., Brom, C.: Beyond smart objects: Behavior-oriented programming for NPCS in large open worlds. In: Lengyel, E. (ed.) Game Engine Gems 3, pp. 267–280. Boca Raton, FL, USA: CRC Press (2016)
29. Witmer, B.G., Singer, M.J.: Measuring presence in virtual environments: a presence questionnaire. Pres. Teleoper. Virtual Environ. **7**(3), 225–240 (1998). https://doi.org/10.1162/105474698565686