



## CSS for CVR: A Reciprocal Velocity Obstacle-based Crowd Simulation System for Non-Playable Character Movement of Campus Virtual Reality

Yunifa Miftachul Arif<sup>a,\*</sup>, Geovanni Azam Janitra<sup>a</sup>, M. Imamudin<sup>a</sup>, Puspa Miladin Nuraida Safitri A Basid<sup>a</sup>,  
Dedy Kurnia Setiawan<sup>b</sup>

<sup>a</sup> Department of Informatics Engineering, Universitas Islam Negeri Maulana Malik Ibrahim, Lowokwaru, Malang, 65144, Indonesia

<sup>b</sup> Department of Electrical Engineering, Universitas Jember, Sumsari, Jember 68121, Indonesia

Corresponding author: \*yunif4@ti.uin-malang.ac.id

**Abstract**— Along with the development of multimedia technology, an overview of the campus environment for prospective new visitors can be visualized through a 3D virtual environment based on virtual reality. A crowd simulation system is needed to provide an overview of the crowds in campus virtual reality (CVR). The simulation helps make it easier for individuals to predict crowds in certain areas virtually. In this study, we propose using the Reciprocal Velocity Obstacle (RVO) method to support the simulation of Non-Playable Character (NPC) crowds in a visualized virtual environment. RVO implements multi-agent navigation by estimating the possibility of moving without communication between agents and being able to perform collision avoidance. The use of RVO in this study aims to contribute to the collision detection development process for each NPC. The application of RVO is carried out in the development of virtual reality by using Unity3D and Blender asset support tools. The results of this study indicate that the RVO method can be applied in multi-agent navigation. These results were confirmed by the success of the NPC as a simulation agent in selecting routes and independently navigating to avoid collisions between agents without the need for communication. In every simulation, collisions will occur within a set of agents due to high density, which causes complex computations. The development of CSS can help every CVR user experience a virtual environment. In addition, each user can experience a more natural experience with the presence of 3D objects and virtual reality with RVO-based CSS. Furthermore, this research material is expected to be developed from various perspectives and themes related to crowd simulation for games and other simulation media.

**Keywords**— Virtual reality; crowd simulation; reciprocal velocity obstacle; non-playable character.

Manuscript received 19 Jul. 2023; revised 4 Oct. 2023; accepted 16 Nov. 2023. Date of publication 31 May. 2024.  
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



### I. INTRODUCTION

Education is an essential aspect of people's lives [1]. Everyone knows that campuses as educational institutions have high daily mobility [2]. Universitas Islam Negeri Maulana Malik Ibrahim Malang (UIN Malang) is the campus we chose as the case study in this research. The reason is that UIN Malang is a growing campus located in the center of Malang and is flanked by two city streets. This condition causes the traffic of students and employees to increase along with the development of the campus. In addition, introducing the campus environment to new students and people who have never come is especially important. They aim to adapt more quickly to the new campus environment. Therefore, we need a media simulation of the campus environment, which

describes student traffic conditions and crowd simulations. However, the built simulation media must also be able to adapt or resemble natural human behavior in moving, such as how they move and avoid collisions.

Several studies have discussed collision avoidance, for example, to be implemented in robots [3], unmanned aerial vehicles (UAV) [4], and evacuation routes [5], [6]. On the other hand, this research focuses on the movement simulation of a Non-Playable Character (NPC) as a representation of people carrying out activities on campus that are visualized using virtual reality (VR) technology. The goal is to make the created crowd feel more real [7]. A crowd simulation system (CSS) is a solution that forecasts and simulates everyone in a campus environment. Meanwhile, the use of VR can increase the degree of reality felt by players [8]. Consumers are often presented with VR technology as a simulation medium to

improve performance and applications and save costs [9]–[11]. VR is currently emerging as a new medium for use in simulation [12]–[14]. Everyone who uses it can be involved, move, and see from a different perspective [15].

In this study, we designed a simulation with 3D visualization to describe the environment, objects, and buildings in a way that is similar to the real world. One of the problems faced is finding a way for each individual who is programmed to think as generally as possible [16], [17]. In other words, the NPC must be able to avoid collisions and determine the path to take to reach the goal. Multi-agent NPC is the main object in crowd simulation in this research. In this paper, we discuss the problem of multi-agent motion navigation in a campus environment simulation. Each agent is expected to be able to navigate independently without communication and without colliding and encountering other obstacles from agents in the same environment [18]. RVO is one solution to overcome this problem [19]. RVO can be used to simulate scenarios where hundreds of similar agents navigate independently in complex environments. Using this method, it is hoped that the agent can perform realistic actions to produce movements that can avoid collisions.

#### A. Related Works

Research on Velocity Obstacles in multi-agents was conducted by James A. Douthwaite et al. [20]. In their research, they present an analysis of some of the most promising approaches for geometric collision avoidance in multi-agent systems, namely, velocity obstacle (VO), reciprocal velocity obstacle (RVO), hybrid-reciprocal velocity obstacle (HRVO), and optimal reciprocal collision avoidance (ORCAs). Each approach is evaluated based on the increasing agent population and variable sensing assumptions. Wei Li et al. also researched evacuation simulations on underground roads using RVO to assess the effect of the number of exits and different exit locations [6]. In their research, they also try to produce regional evacuation routes. The experimental results show that simulation can increase efficiency during the evacuation process. Jing Gao et al. also researched RVO applied to UAVs for dynamic obstacle avoidance [4]. The simulation results show that the RVO can reduce the UAV's turning angle in a dynamic environment. In addition, Wang et al. also applied RVO to their research for planning emergency evacuation routes in crowded places [5]. The RVO algorithm can simulate automatic obstacle avoidance of crowds during evacuation. In contrast to several similar studies, in this study, we tried to implement RVO for simulating crowds on campus. The simulation uses 3D objects and VR technology, so the results are expected to make players feel more natural and feel in a real environment.

#### B. Paper Structure

This paper starts with an introduction that contains the research background and related works that serve as a reference and differentiator from other studies that have been conducted. Furthermore, the material and methods section discuss crowd simulation, which contains theory and method implementation, while the virtual environment sub-section discusses the assets used in the experimental stage. In the simulation section, we explain the results and analysis of the

experiments that have been carried out. The last section is the conclusion, which contains a summary of the research that has been done.

## II. MATERIALS AND METHODS

In this study, we undertaken to design a simulation experiment containing 3D objects. The goal is to make simulation users feel a real-like environment [21]. In this study, we also completed a crowd simulation that described the condition of UIN Malang when there were many people. Players can virtually feel and see the area around UIN Malang, and we call it Campus Virtual Reality (CVR). They can also see the results of crowd simulations using RVO so that the environmental conditions they encounter feel more natural. In the test scenario, we retrieve data based on the collisions in the primary agent, the average framerate while the simulation is running, and the number of main agents that have made it to the destination point.

#### A. Crowd Simulation

Research on crowd simulation has been carried out in recent years [22]. Crowd simulation research focuses on crowded public places or outdoor areas such as parks, building complexes, and pedestrians. Realistic crowd simulation requires the integration of many aspects [23] and has various objectives, such as queuing evacuation and pedestrian crowding processes [24]. Crowd simulation contains several moving objects or agents that interact with each other [25]. The agent must be able to reflect intelligent path planning through its changing environment. Furthermore, the objects within it must be able to adjust their path [6].

Each NPC has its destination point so that a crowd is formed, which creates a crowd atmosphere. Based on this, it is necessary to have a method to make each agent move towards their destination points. This study discusses real-time navigation for multi-agent motion planning to the destination without colliding with obstacles and other objects. Each NPC can navigate independently without communicating with other NPCs.

RVO is the concept of speed barrier method development [26]. This algorithm allows agents to avoid collisions with other agents without central coordination [5]. RVO is an improved version of VO that allows agent movement free of collisions or oscillations [27]. RVO overcomes the oscillation problem in VO caused by speed resistance by combining reactive properties rather than having to risk collision avoidance [20]. VO lets agents take half the risk of collision avoidance while assuming the other agents involved reciprocate by caring for the other half [28].

RVO implicitly avoids local collisions by assuming other agents have reason to dodge [29]. This concept is expected to produce safe and oscillation-free movements. Fig. 1 shows the oscillations in collision avoidance where two objects, A and B, are simultaneously moving at their respective speeds, namely  $V_A$  and  $V_B$ , so they become  $V_A \in VO_B^A(V_B)$  and  $V_B \in VO_A^B(V_A)$ . If A and B continue, there will be a collision. Collisions can be avoided by agent A changing its speed to  $V_A$ , which is outside the velocity obstacle B ( $V_A \notin VO_B^A(V_B)$ ) and B also changing its speed to  $V_B$  yang, which is outside the velocity obstacle A B ( $V_B \notin VO_A^B(V_A)$ ) [4]. When A and B

have managed to get out of the obstacle course, a new problem occurs changing the speed back to its origin so that objects can reach their destination with the fastest path. Oscillations will happen when this is done.

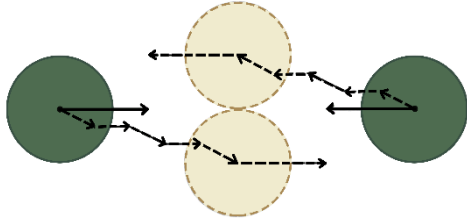


Fig. 1 Oscillations that occur in collision avoidance

RVO eliminates oscillations in obstacle velocity by creating a new obstacle velocity based on the average VAB and other speeds, making the resulting resolution path look smoother. Here is the RVO equation.

$$RVO_B^A(V_B, V_A) = \{V'_A | 2V'_A - V_A \in VO_B^A(V_B)\} \quad (1)$$

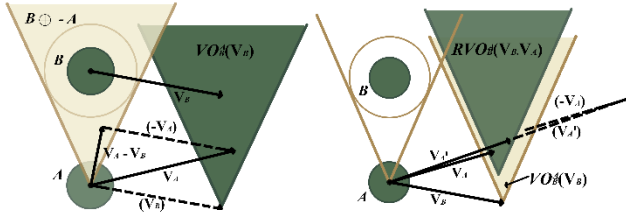


Fig. 2 Velocity obstacle representation (left), reciprocal velocity obstacle representation (right).

Fig. 2 shows that all the  $RVO_B^A(V_B, V_A)$  speeds are the average  $V_A$  and the resistance speed  $B VO_B^A(V_B)$ . The resistance velocity  $VO_B^A$  moves to a peak value at  $\frac{V_A + V_B}{2}$ , geometrically representing  $RVO_B^A$ . Fig. 3 shows the results of the RVO motion, which can avoid collision without oscillation. The velocity that is outside the collection of all RVO regions and closest to the expected velocity is the choice that agents will use based on the valid velocity of RVO.

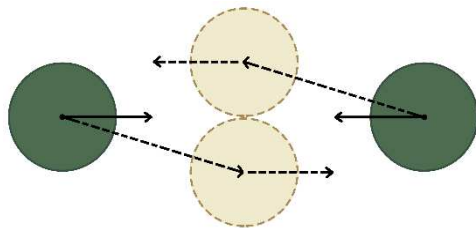


Fig. 3 Agent movement to avoid obstacles using RVO.

The next step is to calculate the RVO algorithm. We assume that  $V_A$  is the current velocity of agent A, and  $V_B$  is agent B. Then, agents A and B assign new velocities to  $V_A'$  and  $V_B'$ , which are outside each RVO. The two agents will not clash or collide and choose the exact part to cross each other using the following equation.

$$V_A \in VO_B^A(V_B) \Leftrightarrow V_B \in VO_A^B(V_A) \quad (2)$$

$$V_A \in VO_B^A(V_B) \Leftrightarrow V_A + U \in VO_B^A(V_B + U) \quad (3)$$

$$V_A \notin VO_B^A(V_B) \wedge V'_A \notin VO_B^A(V_B) \Rightarrow (1-a)V_A + aV'_A \notin VO_B^A(V_B) \text{ for } 0 \leq a \leq 1 \quad (4)$$

$$V'_A \notin RVO_B^A(V_B, V_A) \wedge V'_B \notin RVO_A^B(V_A, V_B) \quad (5)$$

$$V'_A \notin VO_B^A(V'_B) \wedge V'_B \notin VO_A^B(V'_A) \quad (6)$$

$$V'_A \notin RVO_B^A(V_B, V_A) \wedge V'_B \notin RVO_A^B(V_A, V_B) \quad (7)$$

$$RVO_A^B(V_B, V_A) = \{V'_A | 2V'_A - V_A \in VO_B^A(V_B)\} \quad (8)$$

The following equation will be obtained if we implement it in the RVO formula.

$$2V'_A - V_A \notin VO_B^A(V_B) \wedge 2V'_B - V_B \notin VO_A^B(V_A) \quad (9)$$

$$2V'_A - V_A \notin VO_B^A(V_B) \wedge V_A \notin VO_B^A(2V'_B - V_B) \quad (10)$$

$$2V'_A - V_A - V_B \notin VO_B^A(0) \wedge V_A = V_B - 2V'_B \notin VO_B^A(0) \quad (11)$$

$$V'_A - V_B \notin VO_B^A(0) \quad (12)$$

$$V'_A \notin VO_B^A(V'_B) \quad (13)$$

$$V'_A \notin VO_B^A(V'_B) \wedge V'_B \notin VO_A^B(V'_A) \quad (14)$$

Crowd simulation requires that agents can avoid collisions or collisions with other agents. The agent simulation stages in avoiding collisions are shown in Fig. 4 to 7.

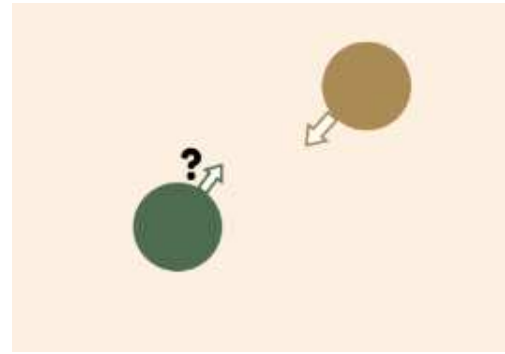


Fig. 4 Agents were facing each other.

Fig. 4 displays two agents in brown and green. The green agent views the yellow agent as an obstacle because the presence of the yellow agent is one of the agents closest to the green agent. While Fig. 5 shows the green agent calculating the outermost area of a collision or collision by considering the area outside the velocity obstacle, which is twice the agent's radius.

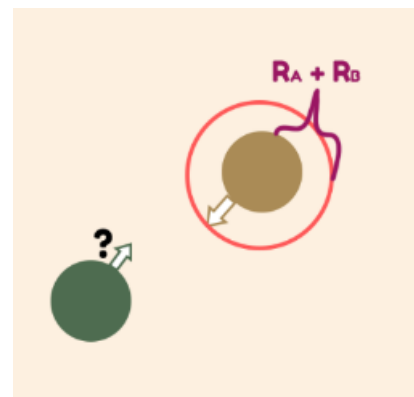


Fig. 5 The green agent calculates the collider area of the yellow agent.

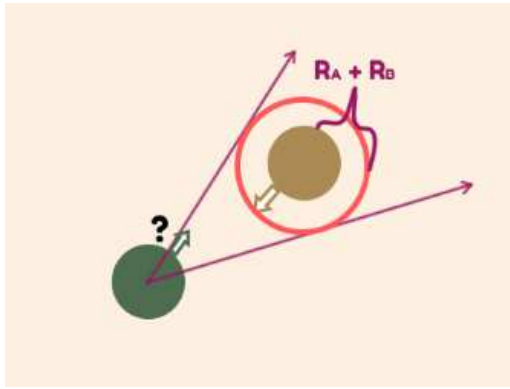


Fig. 6 Velocity obstacle of green agent on the yellow agent

The cone angle centered on the green agent position shown in Fig. 6 is the velocity obstacle area obtained after calculating the set based on all agent velocities and areas that allow collisions. Next, Fig. 7 shows a change in the center of the cone using the RVO algorithm. The moving cone center avoids oscillations and indicates that the two agents will move away from each other.

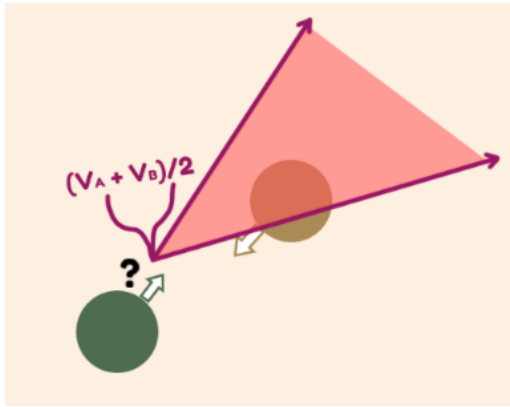


Fig. 7 RVO agent green on agent yellow.

### B. Virtual Environment

In this study, we built a simulation experiment using the Unity game engine. The reason is that Unity has a well-developed system for good graphics, realistic physics simulation, and particle animation [30]. For asset needs, we use the Blender application, which is open-source. The use of Blender helps the 3D workflow because, in this application, there are many facilities such as modeling, texturing, lighting, animation, and finishing. The assets created to make the atmosphere feel more natural, such as roads, buildings, grass, mosques, and others [31]. We designed this simulation to run in an environment that includes a variety of assets, such as roads and buildings. As an agent in the simulation, NPC will run on a flat plane with a NavMesh agent component.

Fig. 8 shows the campus map and various objects supporting the simulation environment, separating objects into static and dynamic [32]. Buildings and grass become static objects that agents cannot pass through, while roads become dynamic objects that help identify a space that can be continuously navigated.

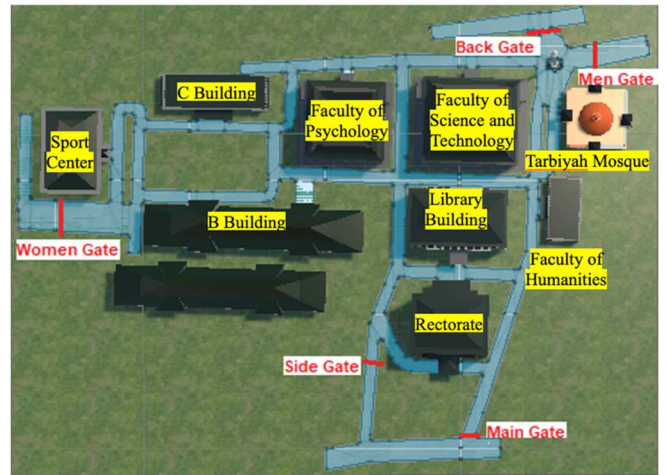


Fig. 8 Campus map.

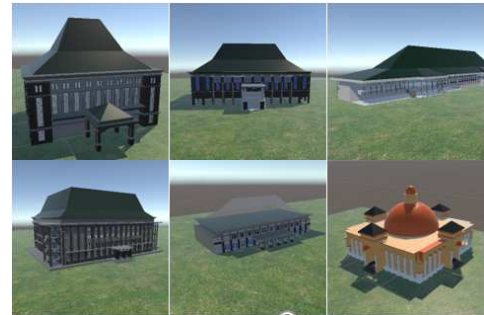


Fig. 9 Destination points in the form of building

Fig. 9 shows the building object, which is each agent's destination point. The system sets the building's destination point randomly. When agents walk to their destination, they encounter other agents with the same destination point or pass each other and try to avoid collisions.

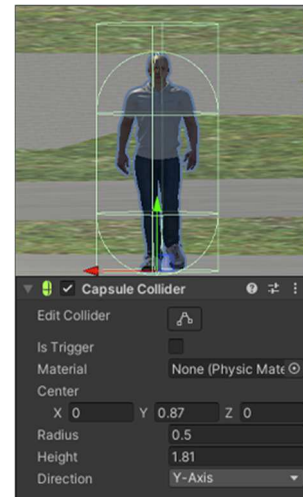


Fig. 10 Agent design

On the other hand, Fig. 10 shows one of several agent designs used in this simulation study. NPC agent designs are made to resemble humans so that the simulation feels more natural with the campus atmosphere. Next, we installed collider components for each agent and static objects that retrieved data on the number of collisions and agents reaching the destination point.



### III. RESULTS AND DISCUSSION

This study uses the Unity3D game engine to visualize the environment and crowd simulation so that it can be displayed

on a VR-supporting device. One of the primary motivations for using RVO is to make the atmosphere more natural in crowd navigation with many NPCs.

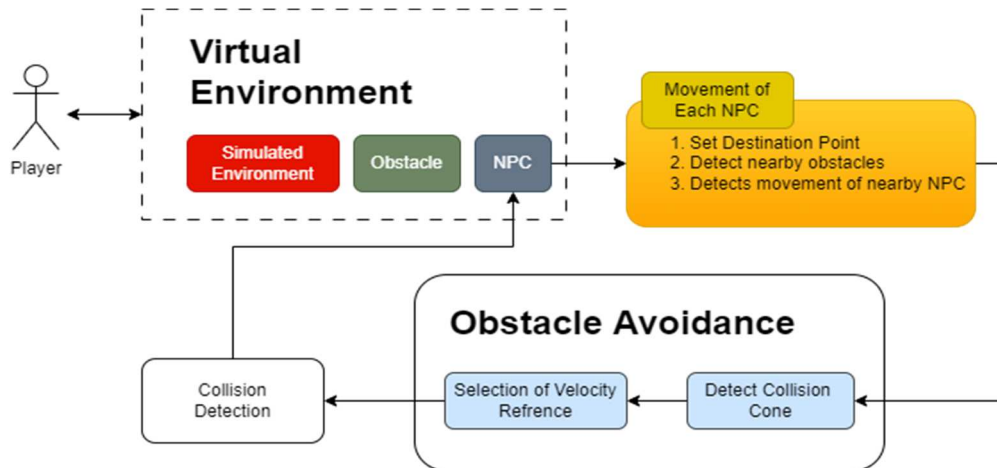


Fig. 11 Proposed CVR

This study shows the proposed CVR work system in Fig. 11. When the player enters the virtual environment. Each NPC walks towards a predetermined destination point. Each NPC will try to avoid collisions between other NPCs and static objects to naturally create a feeling of being in a crowd with VR support. Next, Fig. 12 shows an example of an in-game preview from the player's perspective.

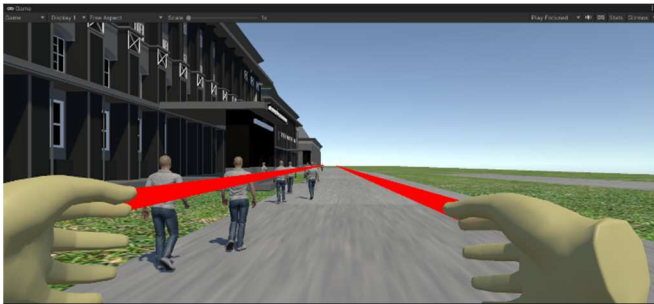


Fig. 12 In-game preview

Fig. 13 shows how NPCs move in virtual reality. The first stage is for each NPC to set its destination point. Then, the NPC tries to avoid collisions with other NPCs or obstacles, and the last is collision detection.

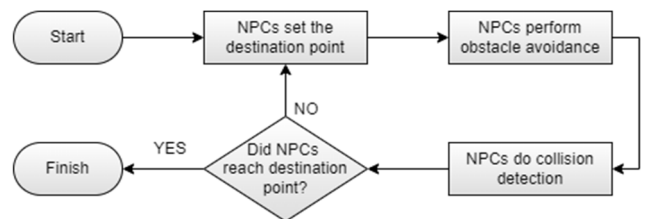


Fig. 13 NPC Flowchart

Fig. 14 shows the collision detection algorithm on the NPC agent. If the distance between two NPCs is less than the sum of their radii, then the NPCs are considered to have collided.

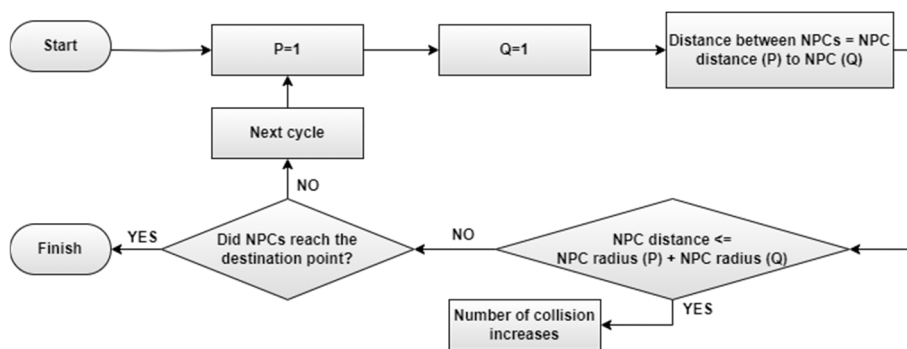


Fig. 14 Collision Detection

In this study, testing was carried out in a virtual environment on the UIN Malang campus, which has five spawn points: the entrance gate, back gate, side gate, female gate and male gate. While the destination point is the buildings scattered in the campus area. Table 1 shows the agent's spawn coordinates.

TABLE I  
SPAWN AGENT COORDINATES

Spawn Point	Coordinate
Main Gate	(10.5; 0.5; -5)
Men Gate	(124.5; 0.5; -48.13)
Back Gate	(137; 0.5; -26.3)
Women Gate	(73.3; 0.5; 118.9)
Side Gate	(28.3; 0.5; 27.9)

While Table 2 shows the agent's finish point coordinates. Each agent's destination point is randomly generated to support the crowd simulation being developed. Next, in our scenario, we design a main agent crowd that runs to its destination point through the crowd simulation. At the same time, the simulation runs for 200 seconds on each route. Fig. 15 shows that each NPC can move towards its target point and not collide with each other a certain number of times.

TABLE II  
FINISH AGENT COORDINATES

Finish Point	Coordinate
Sport Center	(79; 0; 114.2)
Rectorate	(32.5; 0; -8.5)
Library	(59; 4; 1.8)
B Building	(76.8; 0.01; 30.15)
C Building	(104.8; 0; 94.45)
Faculty of Psychology	(128.3; 0; 0.6)
Faculty of Science and Technology	(130; 0; -47.7)
Faculty of Humanities	(86.3; 0; -25.6)
Tarbiyah Mosque	(95; 0; -44.3)



Fig. 15 NPCs woke without collision

Simulation testing in this study was carried out using a computer with an Intel i5 gen 11 processor, NVIDIA GeForce RTX 3050TI, and 8GB of RAM. Based on the number of spawn points and destination points of NPC agents, we ran 45 velocity scenarios from agents. Each test provides a constant for the scenario, namely the agent that enters every second.

Fig. 16 shows the number of collisions with the agent's spawn point at the main gate leading to the nine destination points. The condition in this scenario is that the number of collisions increases as the number of agents increases. The highest collisions were on the route to the Rectorate, with 129 collisions from 200 agents. At the same time, the lowest point is at the Humanities Building, with 42 collisions from the same total agent. While Fig. 17 shows the average framerate when the simulation runs with the main gate spawn point. Based on this figure, the average framerate of 9 points tends to decrease as more agents are added. However, this decline is seen in a small amount but with an up-and-down curve. The highest average framerate is the Rectorate reaching 97 FPS, and the lowest is B Building, with 49 FPS.

The test results on the number of agents who reached the destination point from the main gate within a time range of 200 seconds can be seen in Table 3. Based on the table, SC Building has the least number of agents. This happens because the distance between the starting and destination points is farther than the other routes. The Rectorate Building has the most agents because it is closest to the Main Gate.

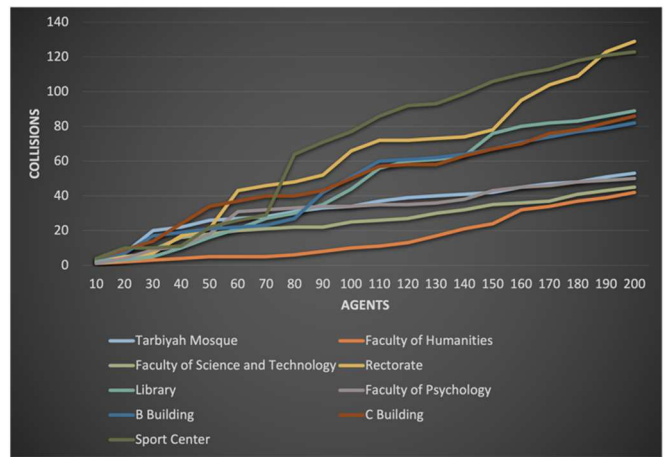


Fig. 16 Total Collision from Main Gate

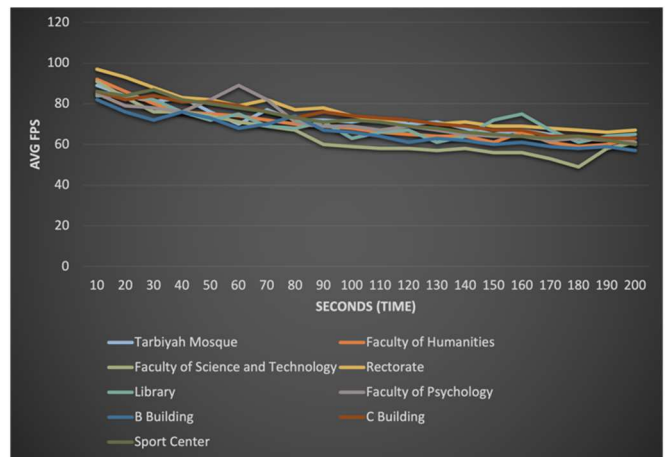


Fig. 17 Average framerate from the Main Gate

TABLE III  
NUMBER OF AGENTS FROM THE MAIN GATE

Destination Point	Number of Agent
Mastar	191
Faculty of Humanities	181
Faculty of Science and Technology	179
Rectorate	195
Library	181
Faculty of Psychology	167
B Building	165
C Building	157
Sports Center	144

Fig. 18 shows the number of collisions between the agent's spawn point at the main gate and the nine destination points. Based on the figure, the total number of collisions tends to increase with the number of agents. The highest number of collisions occurred on the route to the Rectorate, with 58 out of 200 agents. In comparison, the lowest number was at the Faculty of Science and Technology, with 35 of the same total agents.

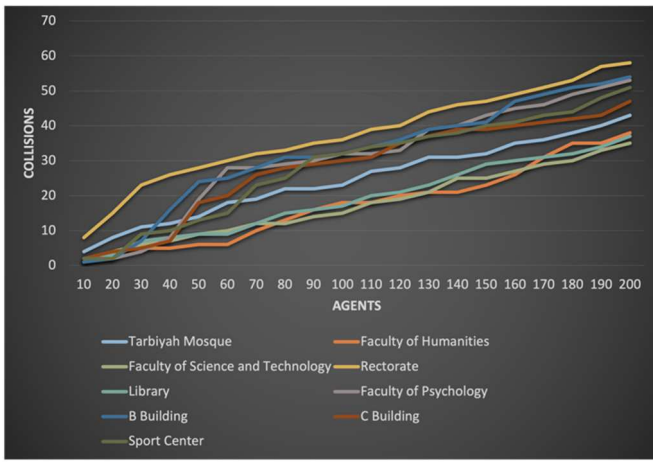


Fig. 18 Total Collision from Men Gate

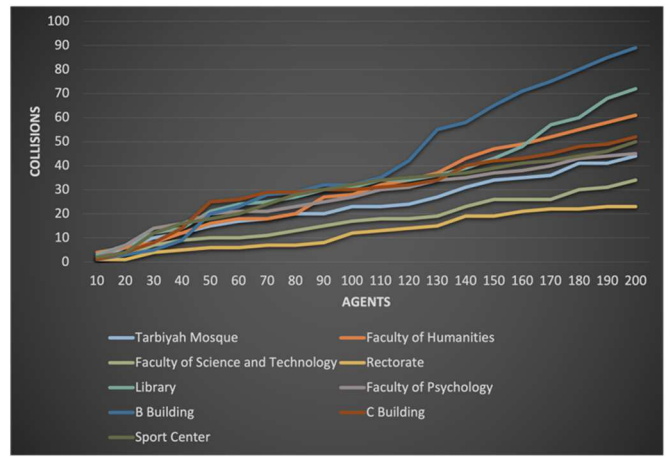


Fig. 20 Total Collision from Back Gate

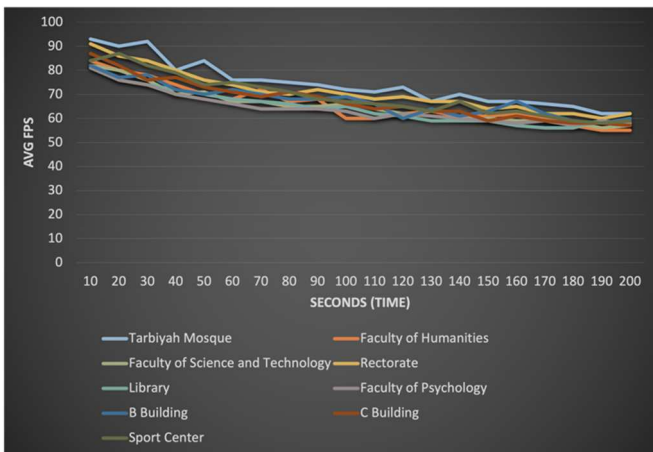


Fig. 19 Average framerate from Men Gate

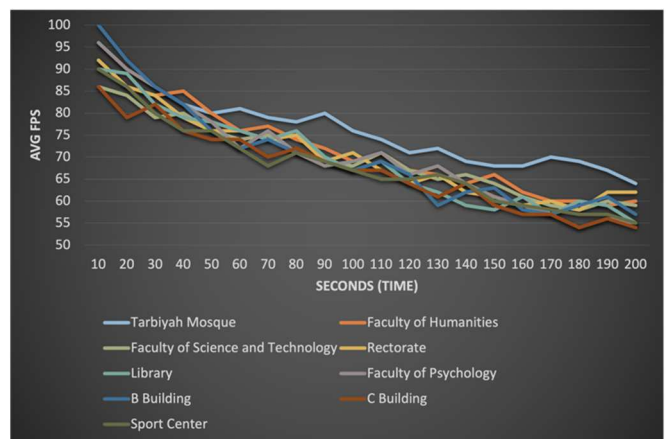


Fig. 21 Average framerate from the Back Gate

Fig. 19 shows the average framerate when the simulation is running with the male gate spawn point. Based on this figure, the average framerate of 9 points tends to decrease as more agents are added. However, the decline is not seen in the up-and-down curve. The highest average frame rate is the Tarbiyah Mosque destination point, which reaches 92 FPS, and the lowest is the Humanities Building, with 55 FPS.

Then, Table 4 shows the results of testing the number of agents who can reach the destination point from the male gate within a specific time range, namely 200 seconds. Building B has fewer agents because the distance between the starting point and destination is farther than the other routes. At the same time, Mastar and Humanities have the same number of agents because they have the closest location to the Men's Gate.

TABLE IV  
NUMBER OF AGENTS FROM THE MEN GATE

Destination Point	Number of Agent
Mastar	201
Faculty of Humanities	201
Faculty of Science and Technology	193
Rectorate	187
Library	222
Faculty of Psychology	201
B Building	162
C Building	184
Sports Center	187

TABLE V  
NUMBER OF AGENTS FROM THE BACK GATE

Destination Point	Number of Agent
Mastar	197
Faculty of Humanities	183
Faculty of Science and Technology	189
Rectorate	178
Library	174
Faculty of Psychology	182
B Building	166
C Building	172
Sports Center	179

The number of collisions that occur with the agent's spawn point at the back gate to the nine destination points can be seen in Fig. 20. Based on the figure, the total collisions that occur also tend to increase with the number of agents. Furthermore, the average framerate when the simulation is running with the back gate spawn point is shown in Fig. 21. Based on the figure, the average framerate of 9 points can be said to decrease as more agents are added. However, this decline is seen in a small amount but with an up-and-down curve. The highest average frame rate is for B buildings, reaching 100 FPS, and the lowest is for C buildings, with 54 FPS. Table 5 shows the results of testing the number of agents who can reach the destination point from the back gate within a time range of 200 seconds. Building B has fewer agents because the distance between the starting point and destination is farther than the other routes. The Tarbiyah Mosque has the most agents because it is closest to the Main Gate.

The number of collisions that occur with the agent's spawn point at the side gate to the nine destination points can be seen in Fig. 22. Based on this, the total collisions tend to increase with the number of agents. The highest total of collisions is on the route to the final destination of the Library Building, with 98 collisions from 200 agents. At the same time, the lowest is the Humanities Building, with 31 collisions from the exact agent total, namely 200. Meanwhile, Fig. 23 shows the average framerate when the simulation runs with the side gate spawn point. Based on Figure 23, the average framerate of 9 points decreases as more agents are added. However, this decline is seen in a small amount but with an up-and-down curve. The highest average frame rate is the destination point of the Tarbiyah Mosque, which reaches 100 FPS, and the lowest is the SC Building, with 46 FPS.

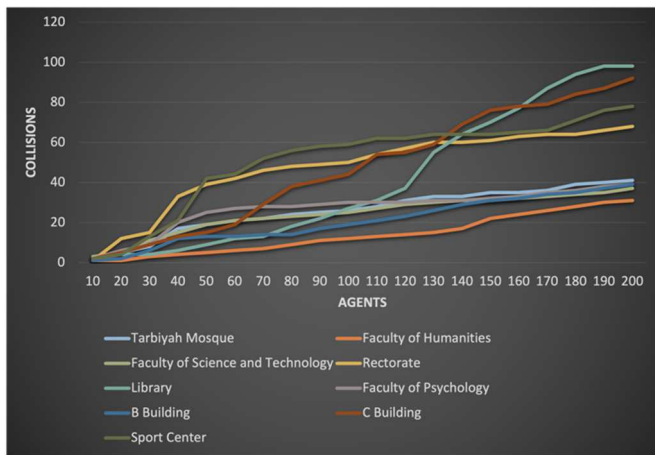


Fig. 22 Total collision from the Side Gate

Table 6 shows the results of testing the number of agents who can reach the destination point from the side gate within a time range of 200 seconds. The library has the most agents because it has the closest location. Meanwhile, agents arrive at other buildings on average at almost the same number because the side gate is in the middle of the UIN Malang campus area. The lowest number of agents arriving at the destination point on this route is the destination for Building C, as it is at the end and passes through a complex crowd point.

TABLE VI  
NUMBER OF AGENTS FROM THE SIDE GATE

Destination Point	Number of Agent
Mastar	192
Faculty of Humanities	190
Faculty of Science and Technology	202
Rectorate	200
Library	252
Faculty of Psychology	205
B Building	200
C Building	182
Sports Center	208

Fig.24 shows the number of collisions occurring with the agent's spawn point at the female gate to the nine destination points. The total number of collisions tends to increase as more agents are added. The highest collisions are on the route to the Humanities Building destination point, with 135 of 200

agents. In comparison, the lowest is the SC Building objective, with 37 collisions from the exact agent total, namely 200. Meanwhile, Fig.25 shows the average framerate when the simulation runs with the female gate spawn point. The average framerate of 9 points decreases as more agents increase. However, this decline is seen in a small amount but with an up-and-down curve. The highest average framerate is the Humanities Building reaching 102 FPS, and the lowest is Building C, with 47 FPS.

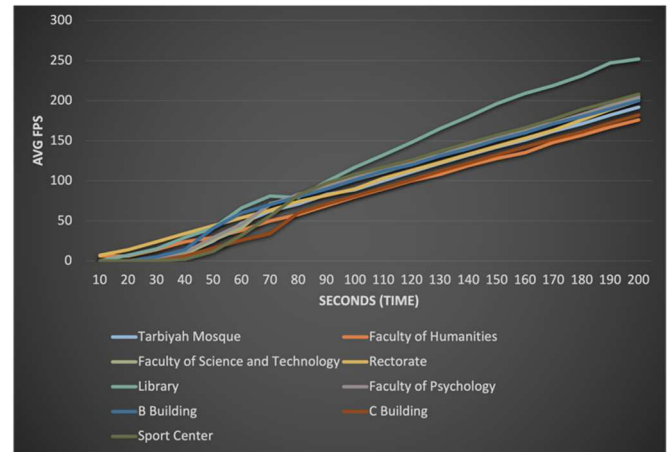


Fig. 23 Average framerate from the Side Gate

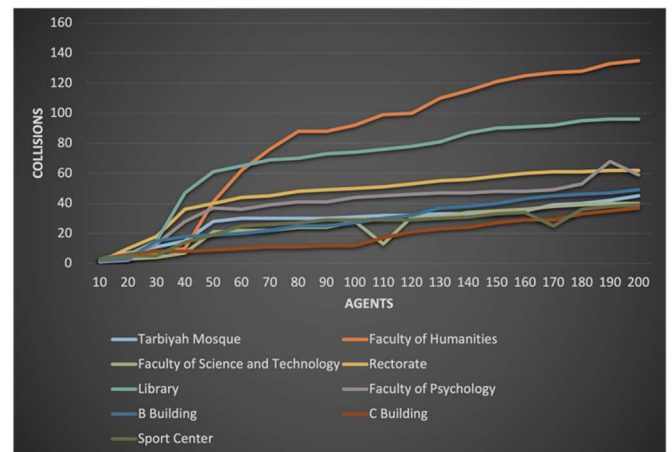


Fig. 24 Total collision from Women Gate

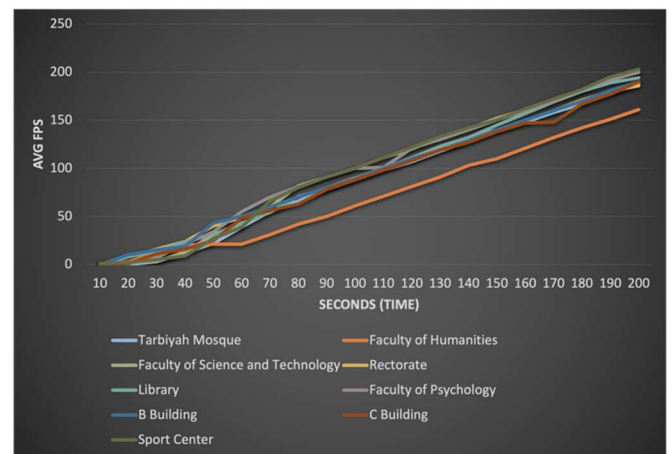


Fig. 25 Average framerate from Women Gate



Table 7 shows the results of testing the number of agents reaching the destination point from the female gate within a specific time range, namely 200 seconds.

TABLE VII  
NUMBER OF AGENTS FROM THE WOMEN GATE

Destination Point	Number of Agent
Mastar	192
Faculty of Humanities	190
Faculty of Science and Technology	202
Rectorate	200
Library	252
Faculty of Psychology	205
B Building	200
C Building	182
Sports Center	208

Based on several test scenarios that have been carried out, RVO can handle several multi-agent navigations. However, too many NPC agents in the simulation will increase the number of collisions and lower the framerate.

#### IV. CONCLUSION

This research explored and implemented the Reciprocal Velocity Obstacle (RVO) method to simulate non-playable character (NPC) crowds within a virtual campus environment, termed Campus Virtual Reality (CVR). The RVO method has demonstrated its effectiveness in enhancing the realism and navigational capability of virtual agents within a simulated crowd by facilitating collision avoidance among NPCs through the autonomous estimation of optimal navigation paths, without necessitating inter-agent communication.

Utilizing tools such as Unity3D and Blender, the RVO method significantly contributed to the development of the collision detection process among NPCs, thereby creating a more immersive and naturally interactive simulation environment for users. Experimental scenarios, focusing on collision data, average framerate, and the navigation of main agents to their destinations, have provided invaluable data, enhancing our understanding of crowd dynamics and autonomous agent navigation within a virtual environment. The test results show that the system can simulate crowds of NPCs in a virtual campus environment where the lowest collision occurred 23 times in the back gate path to the rectorate. In contrast, the highest collision was 35 times found on the women's gate route to the Faculty of Humanities. The average collision that occurs on each route is 34. Furthermore, the number of agents who reached the lowest destination within 200 seconds was 144 who moved from the main gate to the sports center building (SC). Meanwhile, the highest number of agents reaching the destination was 252 from the side gate to the library. The average number of agents arriving at the destination point on each route was 94. Furthermore, the highest average FPS reached 102 on the women's gate route to the Faculty of Humanities. On the other hand, the lowest average FPS reaches 49 on the main gate route to B Building, where the average framerate that occurs on each route is 79.

In the future, further exploration of optimizations and adaptations of the RVO method is essential to ensure its scalability and applicability across various virtual scenarios

and environments. Further research may encompass integrating the RVO method with additional technologies, such as machine-learning algorithms for predictive navigation, and exploring its applicability in contexts beyond campus simulations, such as emergency evacuation simulations and virtual events within more complex urban environments. Exploration of user engagement and experience will also provide a more comprehensive understanding of its implications and potential enhancements in the realm of virtual reality.

#### ACKNOWLEDGMENT

This research was supported by the Multimedia Laboratory, Department of Informatics Engineering, Universitas Islam Negeri Maulana Malik Ibrahim Malang, and International Research Collaboration on Serious Games (IRCSG).

#### REFERENCES

- [1] M. Lövdén, L. Fratiglioni, M. M. Glymour, U. Lindenberger, and E. M. Tucker-Drob, "Education and Cognitive Functioning Across the Life Span," *Psychological Science in the Public Interest*, vol. 21, no. 1, pp. 6–41, Aug. 2020, doi: 10.1177/1529100620920576.
- [2] L. dell'Olio, R. Cordera, A. Ibeas, R. Barreda, B. Alonso, and J. L. Moura, "A methodology based on parking policy to promote sustainable mobility in college campuses," *Transport Policy*, vol. 80, pp. 148–156, Aug. 2019, doi: 10.1016/j.tranpol.2018.03.012.
- [3] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "Independent Navigation of Multiple Mobile Robots with Hybrid Reciprocal Velocity Obstacles," in *The 2009 IEEE/RSJ International Conference on Robots and Intelligent Systems.*, IEEE, 2009, pp. 5917–5922.
- [4] J. Gao, H. Zhang, L. Tan, and X. Ren, "UAV dynamic obstacle avoidance based on improved reciprocal velocity obstacle," *Journal of Physics: Conference Series*, vol. 2216, no. 1, p. 012014, Mar. 2022, doi: 10.1088/1742-6596/2216/1/012014.
- [5] W. Wang, H. Xue, R. Zhou, X. Chen, and Z. Wang, "Simulation algorithm for crowd evacuation in complex situations based on RVO algorithm," *Journal of Physics: Conference Series*, vol. 1861, no. 1, p. 012035, Mar. 2021, doi: 10.1088/1742-6596/1861/1/012035.
- [6] W. Li, Z. Chen, J. Xu, W. Wang, and P. Zhang, "Research on evacuation simulation of underground commercial street based on reciprocal velocity obstacle model," *Journal of Asian Architecture and Building Engineering*, vol. 21, no. 1, pp. 22–33, Nov. 2020, doi:10.1080/13467581.2020.1838289.
- [7] M. Diamanti and H. H. Vilhjálmsson, "Extending the munge crowd simulation framework," *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents*, Sep. 2022, doi:10.1145/3514197.3549698.
- [8] Y. Jang and E. Park, "An adoption model for virtual reality games: The roles of presence and enjoyment," *Telematics and Informatics*, vol. 42, p. 101239, Sep. 2019, doi: 10.1016/j.tele.2019.101239.
- [9] F. Bellalouna, "VR Based Design Process of Industrial Products," *International Conference on Competitive Manufacturing*, no. November, pp. 240–245, 2019.
- [10] J. Adri, F. Prasetya, E. Tasrif, and M. Anwar, "Effectiveness of Using Virtual Reality Media for Students' Knowledge and Practice Skills in Practical Learning," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 3, pp. 688–694, 2023, [Online]. Available: www.joiv.org/index.php/joiv
- [11] S. Sukaridhoto, E. D. Fajrianti, A. Liesvarastranta Haz, R. Putri, N. Budiarti, and L. Agustien, "Implementation of Virtual Fiber Optic Module Using Virtual Reality for Vocational Telecommunications Students," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 2, pp. 356–362, 2023, [Online]. Available: www.joiv.org/index.php/joiv
- [12] J. Pottle, "Virtual reality and the transformation of medical education," *Future Healthcare Journal*, vol. 6, no. 3, pp. 181–185, Oct. 2019, doi:10.7861/fhj.2019-0036.
- [13] H. Nurhayati and Y. M. Arif, "Math-VR: Mathematics Serious Game for Madrasah Students using Combination of Virtual Reality and Ambient Intelligence," *IJACSA International Journal of Advanced*

- Computer Science and Applications*, vol. 14, no. 5, p. 2023, [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [14] “Learning Material Selection for Metaverse-Based Mathematics Pedagogy Media Using Multi-Criteria Recommender System,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 6, pp. 541–551, Dec. 2022, doi: 10.22266/ijies2022.1231.48.
- [15] J. M. Zheng, K. W. Chan, and I. Gibson, “Virtual reality,” *IEEE Potentials*, vol. 17, no. 2, pp. 20–23, 1998, doi: 10.1109/45.666641.
- [16] R. Stern *et al.*, “Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks,” in *Proceedings of the Twelfth International Symposium on Combinatorial Search (SoCS 2019)*, 2019, pp. 151–158. [Online]. Available: [www.aaii.org](http://www.aaii.org)
- [17] “An Artificial Neural Network-Based Finite State Machine for Adaptive Scenario Selection in Serious Game,” *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 488–500, Oct. 2023, doi: 10.22266/ijies2023.1031.42.
- [18] J. van den Berg, Ming Lin, and D. Manocha, “Reciprocal Velocity Obstacles for real-time multi-agent navigation,” 2008 IEEE International Conference on Robotics and Automation, May 2008, doi:10.1109/robot.2008.4543489.
- [19] S. H. Arul and D. Manocha, “V-RVO: Decentralized Multi-Agent Collision Avoidance using Voronoi Diagrams and Reciprocal Velocity Obstacles,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2021, doi:10.1109/iros51168.2021.9636618.
- [20] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, “Velocity Obstacle Approaches for Multi-Agent Collision Avoidance,” *Unmanned Systems*, vol. 07, no. 01, pp. 55–64, Jan. 2019, doi:10.1142/s2301385019400065.
- [21] M. Santos Pessoa de Melo, J. Gomes da Silva Neto, P. Jorge Lima da Silva, J. M. X. Natario Teixeira, and V. Teichrieb, “Analysis and Comparison of Robotics 3D Simulators,” 2019 21st Symposium on Virtual and Augmented Reality (SVR), Oct. 2019, doi:10.1109/svr.2019.00049.
- [22] S. Yang, T. Li, X. Gong, B. Peng, and J. Hu, “A review on crowd simulation and modeling,” *Graphical Models*, vol. 111, p. 101081, Sep. 2020, doi: 10.1016/j.gmod.2020.101081.
- [23] M. Xu *et al.*, “Emotion-Based Crowd Simulation Model Based on Physical Strength Consumption for Emergency Scenarios,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6977–6991, Nov. 2021, doi: 10.1109/tits.2020.3000607.
- [24] F. Muhammad, S. Juniastuti, S. M. Susiki Nugroho, and M. Hariadi, “Crowds Evacuation Simulation on Heterogeneous Agent Using Agent-Based Reciprocal Velocity Obstacle,” 2018 International Seminar on Intelligent Technology and Its Applications (ISITIA), Aug. 2018, doi: 10.1109/isitia.2018.8711117.
- [25] T. Fraichard and V. Levesy, “From Crowd Simulation to Robot Navigation in Crowds,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 729–735, Apr. 2020, doi: 10.1109/ra.2020.2965032.
- [26] T. Xu, S. Zhang, Z. Jiang, Z. Liu, and H. Cheng, “Collision Avoidance of High-Speed Obstacles for Mobile Robots via Maximum-Speed Aware Velocity Obstacle Method,” *IEEE Access*, vol. 8, pp. 138493–138507, 2020, doi: 10.1109/access.2020.3012513.
- [27] S. Juniastuti, M. Fachri, S. M. S. Nugroho, and M. Hariadi, “Crowd navigation using leader-follower algorithm based Reciprocal Velocity Obstacles,” 2016 International Symposium on Electronics and Smart Devices (ISESD), Nov. 2016, doi: 10.1109/isesd.2016.7886709.
- [28] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, “The Hybrid Reciprocal Velocity Obstacle,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, Aug. 2011, doi: 10.1109/tro.2011.2120810.
- [29] S. Juniastuti, M. Fachri, F. Nugroho, S. M. S. Nugroho, and M. Hariadi, “Crowd evacuation navigation for evasive maneuver of brownian based dynamic obstacles using reciprocal velocity obstacles,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2187–2195, Aug. 2022, doi: 10.11591/eei.v11i4.3806.
- [30] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq, “Studying human behavior with virtual reality: The Unity Experiment Framework,” *Behavior Research Methods*, vol. 52, no. 2, pp. 455–463, Apr. 2019, doi: 10.3758/s13428-019-01242-0.
- [31] D. R. Anamisa, M. Yusuf, F. A. Mufarroha, and N. Rohmah, “Design of Virtual Reality Application for Taharah Using 3D Blender,” *Journal of Physics: Conference Series*, vol. 1569, no. 2, p. 022071, Jul. 2020, doi: 10.1088/1742-6596/1569/2/022071.
- [32] M. Aleksandrov, D. J. Heslop, and S. Zlatanova, “3D Indoor Environment Abstraction for Crowd Simulations in Complex Buildings,” *Buildings*, vol. 11, no. 10, p. 445, Sep. 2021, doi:10.3390/buildings11100445.