Jurnal Mantik, 5 (4) (2022) 2117-2126

Published by:Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik/index

Optimization of Virtual Reality-Based Flood Simulation by Manipulating Particle System

David Fahmi Abdillah[1], Achmad Basuki[2], Tri Harsono[3]

[1,2,3]Teknologi Informatika, Pascasarjana Politeknik Elektronika Negeri Surabaya

Email: davidfahmi18@gmail.com

A R T I C L E I N F O

Article history:

Received: Nov 10, 2021
Revised: Jan 22, 2022
Accepted: Feb 30, 2022

Keywords:

Virtual Reality,
Flood,
Cybersickness,
Simulation

A B S T R A C T

VR (Virtual reality) is a technology that is gaining in popularity and many fields have used it in the applications they have developed. VR is the best tool to use as a learning medium and simulations that previously could not be done due to a lack of qualified technology, such as natural disaster simulations where users can feel like one of the victims who are affected by natural disasters and can feel it directly through simulations. the. However, this technology can also have side effects when using it, such as discomfort, dizziness to nausea. This is commonly known as cybersickness. To reduce the possibility of cybersickness, it is necessary to optimize the application by keeping the frame rate at least more than 60-90 fps. Particles are one of the systems that can affect the performance frame rate if not optimized by manipulating their emission for each system. This paper will explain how to optimize VR applications and analyze experiments involving FPS optimization in VR applications.

Copyright © 2021 Jurnal Mantik.
All rights reserved.

1. Introduction

VR (Virtual Reality) is one of the most disruptive technological developments in various fields. This technology has removed any limitations that previously could not be done or achieved. Many new ideas have sprung up in creating or developing a product or invention that can improve that quality. [1] Many fields have benefited from the existence of VR, such as the field of gaming which uses VR to make games more interesting and immersive for players, the field of industry that can create a work environment, and also the field of education that uses VR. for new media educational methods by utilizing VR simulations so that students can be actively involved in the material provided.

The development of VR technology is also inseparable from the emergence of problems from side effects such as cybersickness which has symptoms such as discomfort to nausea and dizziness when using VR applications for a long time. This can reduce interest in VR in gaming, education, and health applications. One way to reduce the possibility of cybersickness is to optimize the performance of the application to reduce flicker and lag when using VR applications. [2]

Frame rate is one of the things that is easier to notice when playing the game. When the frame rate is low, it will make the game look lagging. The lag will look worse in VR because of the way VR work by putting a headgear and visualizing the virtual environment from the headgear. When the environment got lagging, our bodies and our eyes will feel a difference between what we see and what we feel. Usually, that will make people feel dizzy and nauseous. Particles if one of the aspects that can affect frame rate performance because the system will keep emitting particles which makes the application do some extra work. By optimizing the particles, we can find the balance between a good frame rate and a good visual environment to make the player feel immersive and enjoy the game.

The research that will be discussed in this paper is how we simulate rain and our methods to optimize it and find the best performance to minimize cybersickness on users. By testing players and providing feedback from players, we can find data about cybersickness.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

2. Methodology

2.1 General System Description

The environmental simulation that will be used in this experiment is modeling a residential environment like a real-life environment.

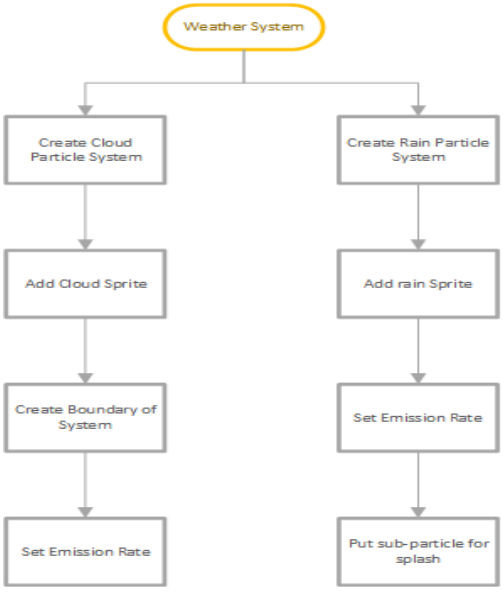

Fig 1. Flowchart Weather System

Flood is one of the important aspects of this simulation. To provide a flood disaster situation, two things are needed, namely rain which is the source of the disaster, and a simulation of rising water in housing. In Unity, there is a feature that can provide rain simulations, namely by using a particle system. A particle system helps to create a large number of objects in a short time in an efficient way.

2.2 Flood Simulation System

In developing a flood simulation, 4 stages are passed in working on a flood simulation in this journal, which can be seen in the diagram below.

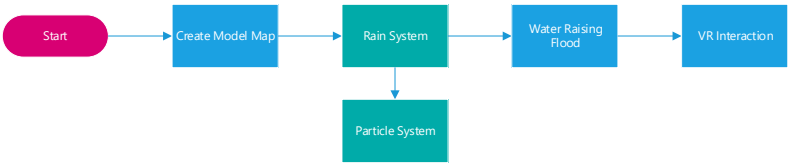

Fig 2. Flowchart Flood Simulation System

The map model in the flood simulation has a background in a residential area in an urban area. The housing does not have sewers or drainage areas, increasing the possibility of flooding. In this simulation scenario, the player will be in one of the houses in the housing, the weather in the simulation is rainy.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Fig 3. View From Above of Simulation Map

When simulating rain, three aspects need to be made to simulate rain realistically. First, simulating clouds for rain, simulating water droplets from clouds, then splashing when water droplets collide with the ground. To model the cloud, create sprites for the cloud for presenters in a particle system. We can use an image editor to create sprites. In this experiment, the sprites we use for the cloud can be found in standard assets by Unity.

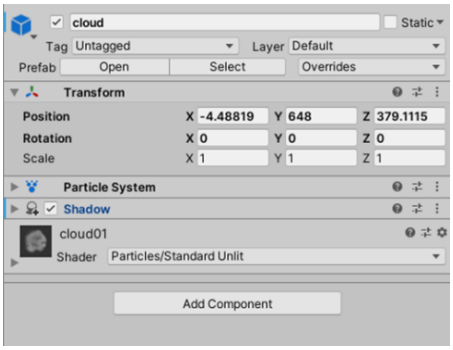

Fig 4. Particle System on Unity

The component particle system has several variables that can be adjusted to provide freedom in providing particle behavior. This variable can be accessed in the module on the particle system component.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

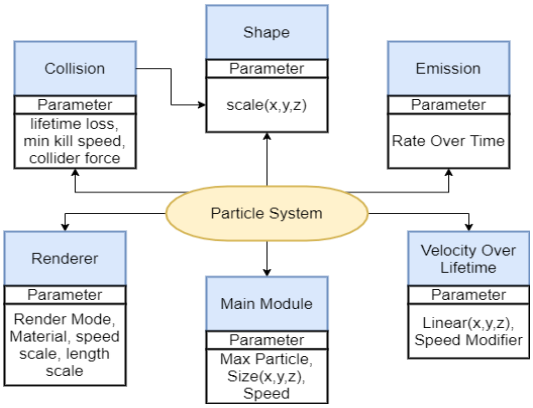

Fig 5. Diagram of Particle System

The main module contains global properties that can affect all particles in the Particle System. The min module is usually used to set the initial state of the particle when it is created.

a. Properties to note are:
b. Duration: how long the system runs per cycle.
c. Starting Lifetime: initial life for particles in the system
d. Starting speed: particle system speed
e. Start size: particle size in the system
f. Particle max: the maximum particle that can exist in the system.

When using a particle system, developers need to know how many particles there can be in one particle system which can affect application performance because the more particles, the harder the GPU works and can result in a decrease in frame rate. By adjusting the lifetime of the particle, and the max particle in the particle system, we can control how many particles can exist at the same time and how long until the current particle dies.

The emission module is a module that is used to regulate particle emissions that will be created by the system. A particle that will be emitted by the system can be controlled by the speed, position between particles. The system cannot emit particles above the max particle set in the main module.

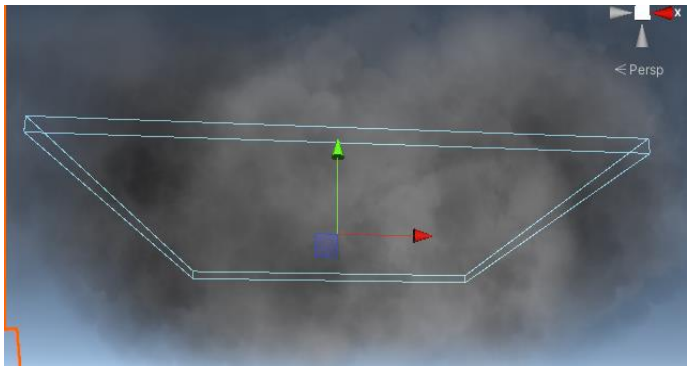

Fig 6. Cloud Particle System

By using the particle cloud system, the simulation can work continuously and the Particle cloud will not radiate out of the box that has been set.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

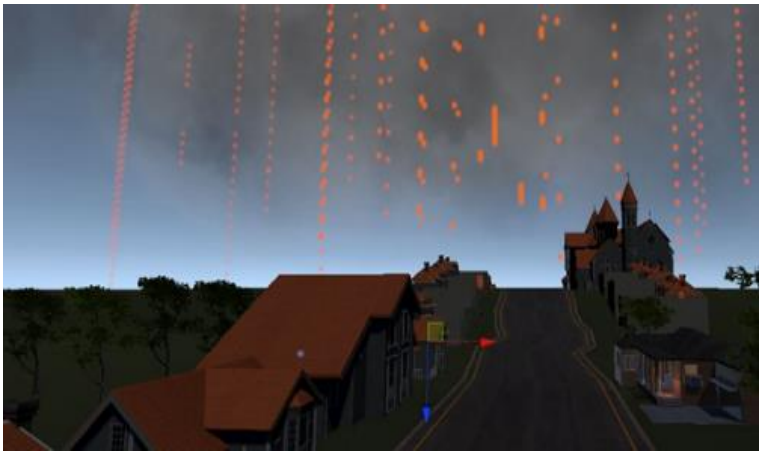

Fig 7. Rain on simulation

When simulating rain, to give a realistic impression of rain, we need to see details when raindrops collide with an object on the earth. When raindrops collide with objects on earth, they will splash and cause ripples.

Rain sprinkling has a different particle system from rain particles. That's because rain and splash have different behavior. However, rain splashes will be created due to the collision of rain particles between raindrops and the earth's ground.

The Collision module can be used to give a particle behavior when it collides with another object in the scene. In this simulation, colliders are used to provide behavior when rain particles collide with objects at the scene. It can regulate behavior when particles collide, whether to bounce, dampen, or disappear immediately. Using collision so that it can be used to emit new particles from other particle systems using the sub emitter module.

In simulating rising water, in this simulation, we will use Shader to simulate water. Shader is a feature in Unity which is a script containing mathematical calculations and algorithms to calculate the color of each pixel based on the lighting input and material configuration. By using the shader feature we can simulate water by using the sin function as a water wave.

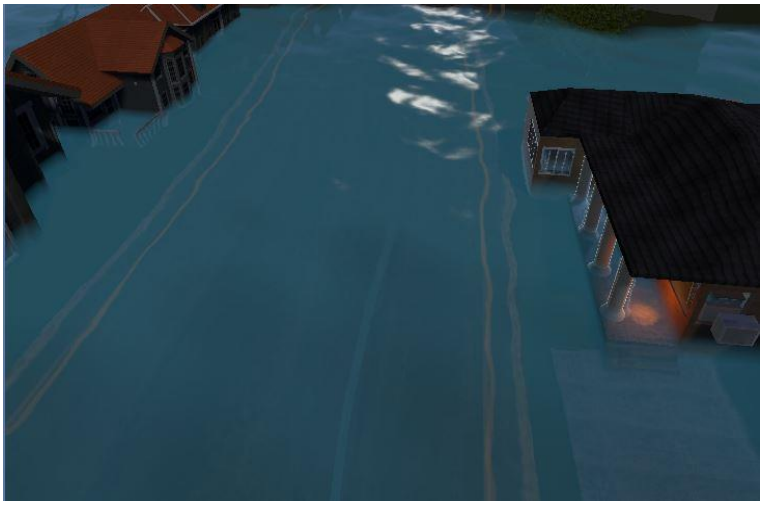

Fig 8. Flood on Simulation

By adjusting the transparency of the object's material, we can give the water a translucent effect. The water will be adjusted to rise as the simulation progresses. Players are expected to complete the given objective before the water rises at a certain point.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

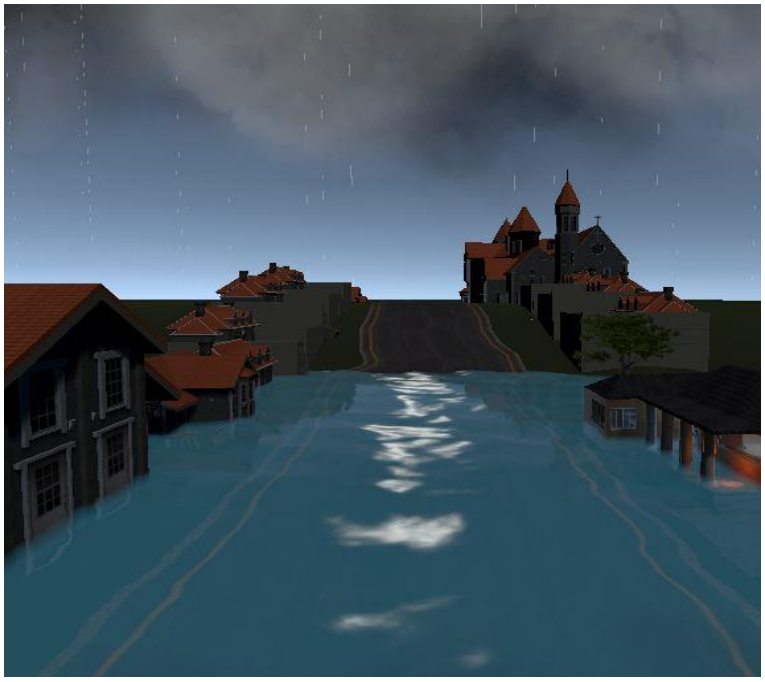

Fig 9. preview of simulation

3. Result and Discussion

The purpose of this experiment is to find the highest particle with the best emission rate for max. The expected result is to find a max particle and emission rate which has an average frame rate above 90 fps and a minimum frame rate above 60 fps. The average fps shows how stable the frame rate is when using the adjusted max particle and emission levels, and the minimum fps shows how much the frame drops.

In a rain particle system, the variables that govern how dense the rain is is the maximum particle and particle emission per second by the system. The properties that need to be considered are the particle max in the main module and the emission rate in the emission module.

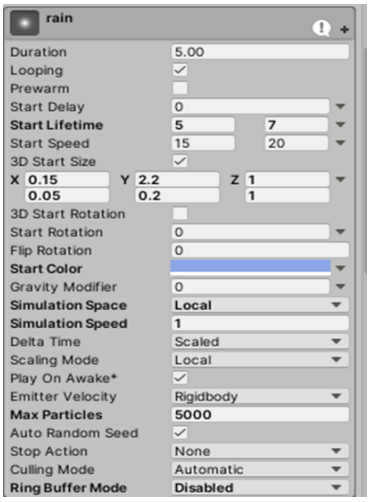

Fig 10. Rain Particle System

The age of the particle in the rain particle system is set between 5-7 seconds, considering that the rain particle will stay alive until it hits the ground. And the particle speed is about 15-20 units per second.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

This experiment was carried out by simulating for approximately 30 seconds, then taking the average frame rate and the minimum frame rate. Particle max and emission rate are independent variables in this experiment, it will produce how many current particles exist at the same time in the rain particle system. In determining the emission rate in this experiment, the max particle rate will be taken by calculating: EmmisionRate=MaxParticle×p/100 Where p is the rate to be set. The p values determined in this experiment are 10, 20, 25, and 50.

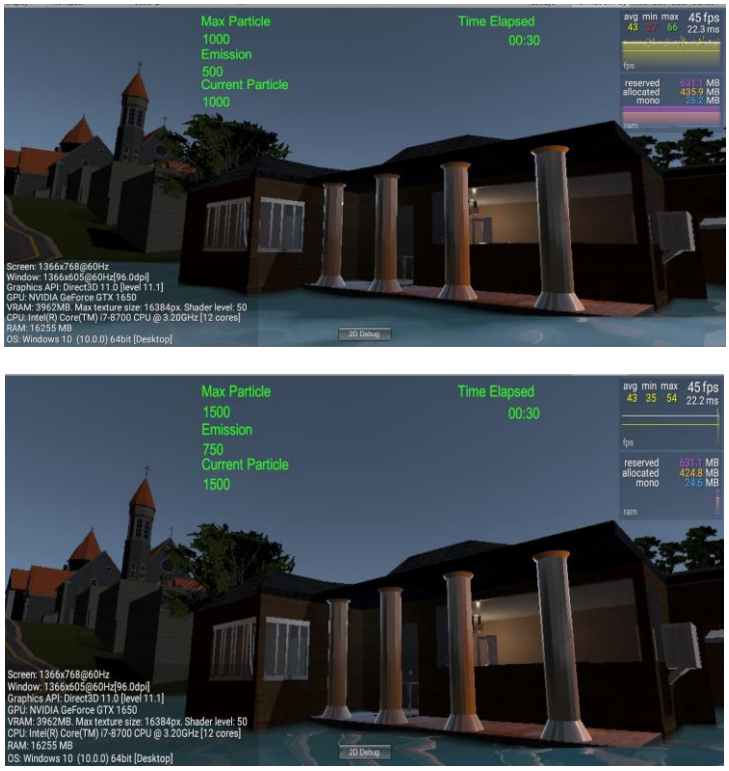

Fig 11. UI for Performance experiment

Graphy is a Unity Asset that can be downloaded at the unity asset store to display the game frame rate. This will be used to show the simulated frame rate when using rain simulation. The UI will display the Max Particle count, emission rate, and the number of currently active particles.

The experiment will be divided into four stages, where each stage has the same max particle number and each stage has four experimental stages that change the emission rate by percentage. The data will be extracted and become a table of observations. [8]

TABLE 1
DATA TABLE OF EXPERIMENT 1

Max Particle	Emission	Avg FPS	Min FPS	Avg Particle
1000	100	116	73	548
	200	111	68	929
	250	112	51	947
	500	103	62	983
1500	150	114	69	823
	300	108	61	1394
	375	106	68	1424
	750	94	59	1475

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Max Particle	Emission	Avg FPS	Min FPS	Avg Particle
3000	300	109	52	1649
	600	96	49	2793
	750	89	46	2846
	1500	76	44	2950
5000	500	103	47	2746
	1000	81	33	4663
	1250	77	33	4748
	2500	59	32	4917

Table 1 shows the average frame rate, minimum frame rate, and particle rate that occurred in the 30-second experiment. Of the four phases, each phase has four experiments, totaling 16 experiments, with the max particle variable used in the experiment, the largest max particle, and the emission rate, the more particles will be made and will reduce performance. simulation on the aspect of the frame rate. Particle max is the maximum number of particles present in one system, and emission is the emission rate of particles from the particle system per second. The average FPS table shows the average FPS the simulation had at run time. And the minimum FPS represents the lowest fps while the experiment is running which indicates how much of a decrease in the simulation. This shows that the particle average below 1500 has an average frame rate above 90 and a minimum frame rate above 60 except for the first trial of the third stage and the second, third, and fourth stages which have a minimum frame rate of below 60.

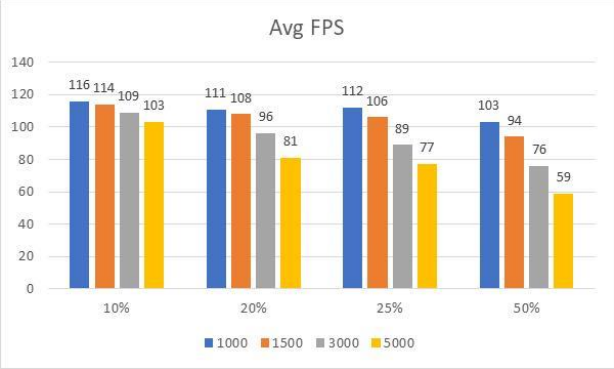

Fig 12. Average FPS Graph

In the graph above, we can see that the larger the emitted particles, the lower the FPS. This is because the more particles there are, the bigger the particles process and the slower the application.

The interesting thing that happens in this experiment is that when the emission rate is above 25% of the particle max, a phenomenon will occur where the rain particles will stop for a moment out of the system and start re-emitting the particles. This will be seen more clearly when the emission level is around 50%. The more time passes, the more stable it will be, but the heavy rain will start to thin out for a while and will start to get thicker again. It will happen continuously.

This phenomenon is related to the age of the particle, and the behavior of the rain when it collides with the ground. It can be seen that in the picture above the number of particles remains at the maximum value, and it is known that the rain particles have a humidity value of 1 in the collider module. That is, rain particles continue to stick to the ground until their useful life is up.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

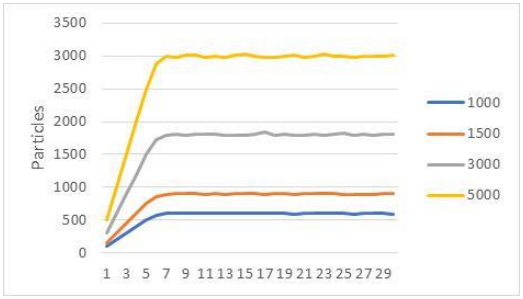

Fig 13. Graph of first Experiment

The picture above is a graph of the change in particle size that lived in the first part of the first experiment. The maximum particle value is 1000 and the transmitter rate value is 100. After 6 seconds, the particle rate remains stable at around 600. This is related to the magnitude of the emission rate and the balanced particle life, where the System continues to emit 100 particles per second, and each particle has a lifetime of approx. 5-7 seconds.

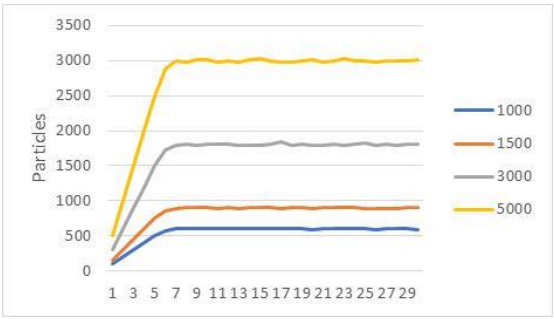

Fig 14. Graph of second experiment

The picture above is a graph of the change in the size of the particle that lives in the first fourth part of the experiment. In this experiment, the maximum particle value is 1000 and the emission rate is 500. The number of particles live directly at the maximum particle value in about 3 seconds and will be maintained at the max value. This is because the emission rate and lifetime are not balanced, thus making the system stop emitting particles until the particle's lifetime runs out. This is what makes the phenomenon of particle rain stop emitting for a moment. And it will make the number of particles in the system is at its maximum value.

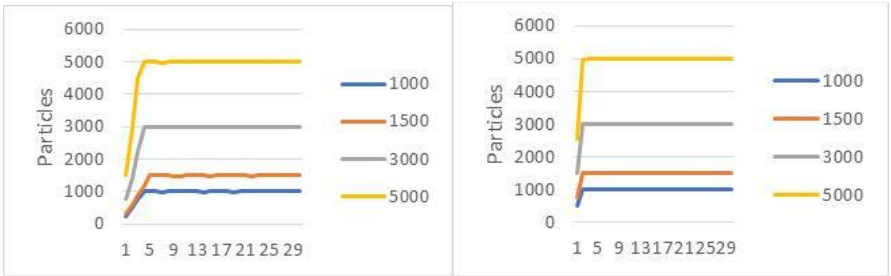

Fig 15. Graph of third Experiment(left) and fourth Experiment (right)

In the graph above is the movement of the particle in the 25% experiment and 50% emission rate. It can be seen that the greater the emission rate, the faster the particle touches the max particle value, resulting in heavier application performance. This needs to be prevented because it can interfere with application performance and the comfort of players in using the application.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

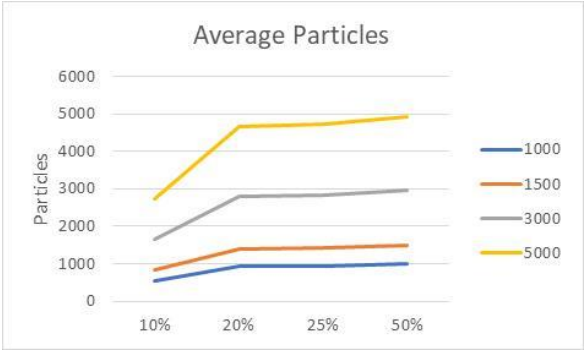

Fig 16. Average Particles

The graph above is the change in the average particle value in each emission rate experiment. Starting from the 20% to 50% trial, there was no significant change compared to the difference between 10% and 20%.

4. Conclusion

The particle system is one of the features in the unit which is very useful for making simulation modeling of natural events such as creating rain, clouds, and splashes of water, and other natural events. And VR is one of the technologies that are very useful for simulation because it makes it easier for players to explore because they feel they are actually in the simulation.

This paper discusses rain simulation using the Particle System in Unity and finds a balance to provide realistic simulations with reasonable frame rates for VR applications by manipulating variables such as particle max, emission rate, and particle lifetime. Particle lifetime affects the particle cycle in the system. Although many variables can affect particle systems such as the collider module, the renderer module can be optimized to improve system performance. From the experiment, it was shown that the best number of particles is around 1000 particles which are following the guidelines, it is said that the best draw call for VR applications is around 500-1000 draw call objects.

Experiments show how important it is for developers to pay attention to the balance of idealism to achieve realistic simulations and stability of simulation performance so that players can feel comfortable using simulations. By finding a balance to provide realistic simulations and stable performance and existing guidelines that help developers to optimize simulation performance especially on the frame rate side to create the best simulation experience especially using VR..

5. References

[1] D. Mishra, "Simulation of Water Inundation Using Virtual Reality Tools for Disaster Study: Opportunity and Challenges," in Indian Institute of Space Science and Technology Trivandrum Kerala .

[2] S. Michalak, "https://software.intel.com," Intel, 3 July 2017. [Online]. Available: https://software.intel.com/en-us/articles/guidelines-for-immersive-virtual-reality-experiences. [Accessed 22 December 2019].

[3] Z. Bingqing and H. Wenfeng, "Game Special Effect Simulation Based on Particle System of Unity3D," IEEE, 2017.

[4] l. S and W. W, "Study on algorithm for fireworks simulation based on particle system," Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, pp. 231-234, 2013.

[5] I. Sabilirrasyad, A. Basuki and Harsono, "PERMODELAN VISUAL TINGKAT KETAKUTAN PADA SIMULASI," JIRE (Jurnal Informatika & Rekayasa Elektronika), vol. 3, no. 1, pp. 30-39, 2020.

[6] J. Bardi, "marxentlabs.com," Marxent 3D Commerce, 7 March 2019. [Online]. Available: https://www.marxentlabs.com/what-is-virtual-reality/. [Accessed 23 December 2019].

[7] D. Simon, K. Nesbitt and E. Nalivaiko, "A Systematic Review of Cybersickness," 2014.

[8] D. F. Abdillah, A. Basuki and T. Harsono, "Visually Realistic Rain Modeling Optimization for VR Application," in International Electronics Symposium (IES), Surabaya, 2020.

Jurnal Mantik is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).