

VR-Based Disaster Rescue and Preparedness Simulator

Naveen S¹, Thejesh N S², Gunasekaran S³

^{1,2,3} Department of Computer Science and Design Engineering, Rajalakshmi Engineering College, Thandalam - 602105, Chennai, Tamilnadu, India

Abstract—Quick Save - Virtual Reality (VR) based disaster response simulation designed to enhance user’s preparedness, awareness, and decision-making during emergency situations. Simulation designed to give people that practice in a safe, digital environment. Built using Unreal Engine 5, this tool improves user preparedness by making them face a realistic emergency. The training are two stages. First stage, user’s is a Survivor. They must escape a destroyed building, avoiding like falling rocks and blocked paths to reach safety. In the second stage, the users become a rescuer, locate and communicate with victims and guide them to the safe zone. A score system is integrated into the user interface (UI) to evaluate player performance. This feedback helps users learn how to stay calm and save lives.

Index Terms—VR Simulation, Disaster Response Training, Interactive Learning, Emergency Preparedness, Rescuer Simulation, Safety Training, Unreal Engine 5, 3D objects Blender

I. INTRODUCTION

When a disaster happens, both natural (like earthquakes and floods) and manmade (like building collapses or fires) everyone gets scared and confused. In that moment, acting fast is the only way to survive. The difference between life and death in these situations is the make fast, smart decision. However, most people do not have the right skills to handle this pressure because they have only learned about safety in a classroom or through simple videos. Virtual reality (VR) is a computer technology that puts you inside a digital environment. Because it feels real, your heart rate might go up, and you fell a little bit of stress. This is good for training. It teaches you how to think clearly even when you are nervous.

Quick Save - VR Based Disaster Rescue and preparedness Simulator, uses this technology to create a better training tool. We built this system using Unreal Engine 5, which allows for very realistic graphics and

physics. The main goal of “Quick Save” is to offer a complete training experience by using unique both perspective approach. This means the user experiences the disaster in two different roles.

1. Playing as a Survivor: First, the user plays as a victim trapped in the disaster. They must focus on saving themselves, navigating through find their own way out.
2. Playing as a Rescuer: Afterward they go back into the building with the right gear to find other victims who are stuck. They have to help the AI victims and lead them to a safe zone.

II. SYSTEM DESIGN

2.1 How the Simulation Works

The “Quick Save” system is designed with a clear structure.

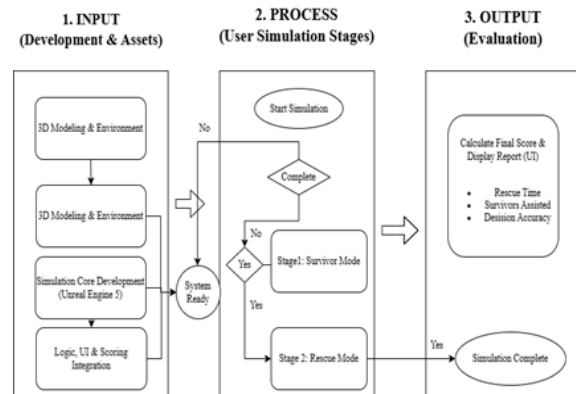


Figure 2.1: VR-Based Disaster Rescue and Preparedness Simulator

The Setup (Input): Before the simulation Starts, build the virtual worlds. This involves creating 3D models, setting up the physics in Unreal Engine 5, and programming the game rules and scoring logic to make the system ready for use.

Action (Process): This is the process in which the user plays. The simulation consists of two significant steps after which we can distinguish:

Survivor Mode: The user tries to escape the calamity.
Rescue Mode: The user is also switching personalities to save other individuals.

The Results (Output) : The system will act as a teacher marking a test when the mission is over. It gives a final grade and it gives out a report card to the user in which Rescue Time, Survivor Time and Decision Accuracy are shown therefore the user is at a position to know the performance that he/she was able to attain.

2.2 Simulation Modules

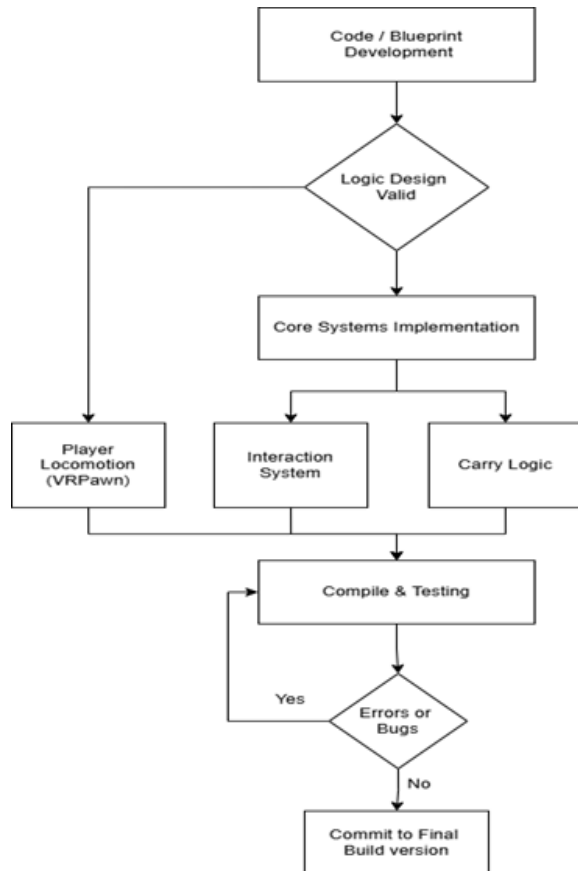


Figure 2.2: Code/Blueprint Development in Simulation Modules.

The game has three modules that operate the dual perspective game play. These are special purpose Components, including the Environment Module, which takes care of the real time physics of the structural collapse; the Survivor Module which concerns the health of the player, and the evasion logic;

and the Rescuer Module which handles AI behaviour and the interaction logic needed to save victims.

Program Blueprint Scripting: The programmer begins with the variables and classes required in the program.

Logic Design Check: The logic should be validated manually, to determine whether it makes sense on paper before writing it in code form. In case the logic is not right (e.g. a survivor with an infinite health), you restart all over.

Implementation: The coding should be divided into three activities:

Locomotion of the player: Writing the code of seamless motion and angular turn.

Interaction System: Coding interaction of hand grabbing of assets.

Carry Logic: The code that is special and makes the victim attached to the player and indicates the speed of movement.

Compile Test: These are the Compiles of Unreal Engine. To test if the code works.

Error or Bugs: In case the victim got stuck in a wall, or the carry mechanic failed to reduce the speed of the flow, the flow returns to the testing stage to be debugged.

Commit to Final Build: When no bugs are detected the code will be stored as the final version to use in the conference demo.

2.3 Simulation Scenario Flow

The simulation begins with the Start Simulation phase, when the system sets up the environment and asks the user to make a choice during Role Selection. Once the user decides whether to play as a Survivor or a Rescuer, the system immediately runs the Execute Fracture event, which causes the building to physically collapse and creates a realistic mess of debris. On this point, the game changes based on the select role: if the user is Survivor, the system turns on the Enable health feature, meaning they must protect themselves from injury while trying on the Reach Safe Zone. When the user is a Rescuer, his or her objective will become to rescue other people, s/he will need to Find AI Victim in a ruins and Guide Victim out of it. The two tracks culminate into the same final step, Calculate Score, whereby the game evaluates the speed and efficiency of the user and rewards him with a final grade.

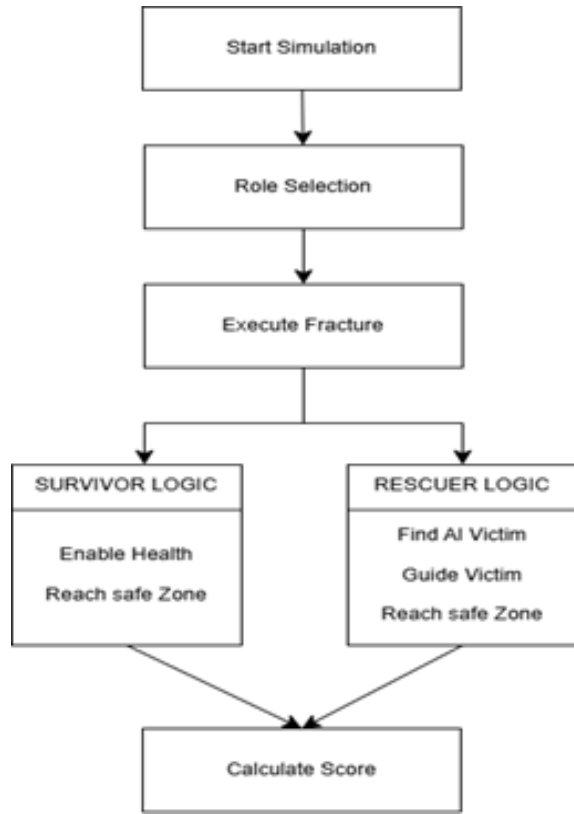


Figure 2.3: Simulation Scenario Flow Diagram.

2.4 Interaction Mechanics

The following sections provide detail of the way the user will manipulate the simulation and interact with the virtual world. The usual Unreal Engine VR tools were used, but had to be customized to meet our requirements.

Movement: With the intention of avoiding the feelings or motion sickness among the users, we have employed a motion system. To move to the ground the user finds the appropriate place and points a laser at it and presses a button to be transported instantly. This is not as uncomfortable as walking in a smooth manner in a video game.

Picking Up Objects: One can pick up the objects by pushing the Grab button on their controllers. They will be able to shift debris such as rocks to clear a way or collect equipment such as flashlights. Necessary things may be kept in a virtual belt on the waist of the user.

Carrying Victims: The user has to walk up to an injured person and press the grip buttons to save a person. This links the victim to the gamer. To make it realistic, the system automatically reduces the walking speed of the gamer by 40 percent when he has somebody being

carried over the shoulder, as an attempt to mimic the heavy load.

User Interface (UI): Rather than numbers floating around the screen we display the game information on a virtual screen.

III. IMPLEMENTATION

3.1 Simulation Environment & Assets.

Unreal Engine 5 was selected due to its features in real time rendering and was used to create the simulation environment.

Assets Creation: Reined buildings and debris Custom 3D models of the reined buildings and debris were made in Blender. These assets were brought as FBX assets into Unreal Engine 5 as imports.

Virtual Reality Setup: The project will be supported with using OpenXR plugins, which will make it compatible with the major VR headsets. (like Oculus).

Player Setup: Upon intonation, we transformed the default " VR Pawn" (the player character) in to the Character Class of the engine. This makes a player not walk through the walls and start being physically involved with the rubble.

Lighting: To achieve the appearance of a scary and real disaster we used Lumine. It is a lighting system that is working with the calculation of the bouncing of light in real time, which provides the appearance of realistic shadows in the dark, collapsed building.

3.2 Destruction Logic

To transport a simple animation, the undertaking uses the physics engine of Unreal Engine, which propels the collapse of the structure.

Geometry Collections: Geometry collections were generated out of normal static meshes. This enables the engine to manage a solid body as a brittle body having internal stress fractures.

Fracture: we utilized the uniform fracture method to pre-slice the building meshes into thousands of smaller, jagged pieces. The pieces remain connected until the simulation triggers the event.

Anchor Fields: To prevent the building from falling immediately, "Anchor Fields" were established at the base of the structure. The collapse is triggered blueprints by disabling these fields, causing the physics engine to calculate gravity and collision for every individual brick in real time.

3.3 AI Navigation

NavMesh: Standard navigation meshes are static. We configured the “NavMesh Bound volume” to runtime generation.

Behavior Trees: The victim intelligence is governed by Behavior Trees. The logic flows through sequence Idle State → Detect Player→

Play animation.

IV. FUTURE ENHANCEMENT

4.1 Maritime Search and Rescue (SAR) Module

Shipwreck Scenarios: We will add a new level where a large ship has broken down or is sinking in the middle of the ocean. The simulation will use realistic water physics, making it hard to move. The player will act as a rescue swimmer or a lifeboat operator.

Saving Drowning Victims: The main goal will be to save passengers who have fallen into the water. Player must quickly throw life jackets or ropes to victims before they drown. They will also need to help people get into life rafts while fighting against strong waves and bad weather.

4.2 Airplane Crash Response

Crash Site Search: This scenario will take place at the site of a plane crash. The area will be a very large place, with wreckage and debris scattered everywhere. The player must search through the broken parts of the plane to find survivors who are stuck inside of the plane.

4.3 Human made and Industrial Disasters

Factory Accidents: The module focuses on accidents caused by machines or chemicals, such as a gas leak in a factory, house or a collapsed mine. Different types of natural disasters, these are caused by human error or equipment failure.

V. CONCLUSION

This research presents the design and implementation of “Quick Save” a natural disaster simulation framework built on Unreal Engine 5. The Project successfully creates a safe, immersive, and highly realistic environment where users can practice lifesaving skills training without the risks associated with real world physical drills. Different traditional training methods, which can be expensive and logistically difficult to organize, the Virtual Reality

system provides an accessible platform for repeated practice in a controlled setting. By separating the environment, player, and artificial intelligence modules, the system achieves high performance without reducing system speed of virtual reality. The successful prototyping of the “Dual Perspective” logic flow proves that consumer grade VR hardware can support modern rescue simulations. Survivors learn crucial evasion and self-preservation techniques, while rescuers practice the stress management and decision-making skills required to save others.

This research confirms that VR is a scalable, lifesaving educational platform that empowers individuals to act confidently in the face of real-world crises.

REFERENCES

- [1] C.Elendu et al., “The impact of simulation-based training in medical education: A review,” *Medicine*, vol. 103, no.27, Jul.2024.
- [2] K.Karamchandani et al., “Emergency airway management outside the operating room: Current evidence and management strategies”, *Anesthesia Analgesia*, vol. 133, pp. 648-662, Jun.2021.
- [3] S.Sharma et al., “Immersive Virtual reality environment of a subway evacuation on a cloud for disaster preparedness and response training”, in *proc.IEEE Symp.Comput. Intell. Human-Like Intell.* Dec. 2024, pp. 1-6.
- [4] S. P. Smith and D. Trenholme, “Rapid prototyping a virtual fire drill environment using computer game technology”, *Fire Saf. J.*, vol. 44, no. 4, pp. 559-569, May 2009.
- [5] S. Ooi, T. Tanimoto, and M. Sano, “Virtual Reality fire disaster training system for improving disaster awareness”, in *Proc. 8th Int. Technol.*, Mar.2019, pp.301-307.
- [6] H. Mitsuhashi et al., “Failure-enhanced evacuation training using a VR-based disaster simulator: A comparative experiment with simulation evacuees”, *Proc.Comput. Sci.*, vol.159, pp. 1670-1679, Jan.2019.
- [7] D. Cho., “Detection of the stress levels from biosignals measured in the virtual reality environment using a kernel-based extreme learning machine”, *Sensors*, vol. 17, no. 10, p. 2435, 2017.

- [8] N. Sharma and T. Gedeon, "Modeling stress recognition in typical virtual environments", in Proc. 7th Int. Conf. pervasive computing technologies for healthcare, 2013, pp. 17-24.
- [9] P. Mickiewicz et al., "The assessment of virtual reality training in antromastoidectomy simulation", *Virtual Reality*, Apr. 2021.
- [10] D. Yang, Y.-K. Wei, F.-S. Xue, X.-M. Deng, and J. Zhi, "Simulation airway management training: Application and looking forward", *J. Anesthesia*, vol. 30, no.2, pp.284-289, Apr.2026.
- [11] A. L. Ramos, J. V. Ferreira, and J. Barcelo, "Model based system engineering: An emerging approach for modern system", *IEEE Trans. System. Man, Cybern., C Appl. Rev.*, vol. 42, pp. 101-111, Jan. 2012.
- [12] S. Jain, B. D. B. Chowdhury, J. M. Mosier, V. Subbian, K. Hughes, and Y.-J. Son, "Design and Development of an Integrated Virtual Reality based training System for Difficult Airway Management ", *IEEE Journal of Translational Engineering in Health and Medicine*, vol.13,2025.
- [13] J. Agarwal and S. Shridevi, "Procedural Content Generation using reinforcement learning for Disaster Evacuation Training in a Virtual 3D Environment", *IEEE Access*, vol.11,2023.
- [14] A. Saidatul, M. Paulraj, S. Yaacob, and M. Yusnita, "Analysis of eeg signals during relaxation and mental stress condition using AR modeling techniques", in Proc. 2011 IEEE International Conference on Control System, Computing and Engineering, 2011.