Colton Sammut

Virtual reality fire evacuation simulation using unity for Xamk campus

Bachelor’s thesis

Bachelor of engineering

Degree programme in game programming

2025

South-Eastern Finland
University of Applied Sciences

Degree title	Bachelor of Engineering
Author(s)	Colton Sammut
Thesis title	Virtual reality fire evacuation simulation using unity for Xamk Campus
Commissioned by	Xamk
Year	2025
Pages	49 pages, 2 pages of appendices
Supervisor(s)	Jani Ahde, Pekka Vilpponen

ABSTRACT

This thesis presents the design, development, and evaluation of a VR fire evacuation simulation for the Xamk Kotka campus that was ecologically valid and immersive. The primary objective was to address the limitations of traditional fire drills. This approach offered campus occupants a flexible, risk-free, and engaging alternative that helps overcome workflow disruptions, infrequent repetition, and limited realism. The study examined technical and user experience challenges in developing a VR system. It aims to balance environmental fidelity with performance on accessible hardware.

The methodology employed a mixed-method approach. The Unity Experiment Framework automatically recorded quantitative data on evacuation durations and response times. The study gathered qualitative data through Likert-scale surveys and participants observations. Fifteen participants underwent three trials: two with assistance and one without. The results demonstrated a 24% decrease in evacuation times and a 40% enhancement in response times, highlighting improved decision-making and knowledge retention. Survey results indicated that participants were confident in evacuation skills and agreed on the simulation's ecological validity and training benefits. However, users reported motion sickness, especially during stair navigation.

The findings presented in this thesis demonstrate that VR-based fire evacuation training can enhance preparedness, spatial awareness, and user engagement while minimising disruption. The study concluded that VR simulations complement traditional drills and suggest further research on skill retention, accessibility, and the integration of adaptive and social features to improve training effectiveness.

Keywords: Thesis, Documentation, VR, Fire Drill, Simulation, Unity 3D, OpenXR, Unity VFX, Unity Experimental Framework, Simulation-based training,

CONTENTS

APPENDICES

1 INTRODUCTION

Safety is crucial in any institution to prevent injuries and damage. Fire outbreaks pose a significant threat to large institutions. These outbreaks are why management takes preventive measures to reduce the impact of such outbreaks. Fire drills are essential to prepare occupants for potential fires. Institution management schedules fire drills for all occupants to practice evacuating, locating fire exits, and measuring response times. As a result, new occupants may remain unaware of fire safety procedures for months, and the lack of realistic threat in traditional drills can lead to less severe participation during evacuations.

This thesis developed and assessed a VR simulation for the Xamk campus in Kotka to address the problem. This tool offered a realistic training experience that enhanced fire safety education by simulating real-life emergencies. Trainees experienced challenging situations, such as navigating through smoke-filled areas or responding to fire alarms, without any actual danger. Conventional drills usually require substantial time, financial resources, and physical infrastructure. By eliminating the need for physical equipment and supplies, VR reduced the costs associated with equipment procurement and maintenance. Furthermore, VR training offered greater flexibility than traditional drills, as participants could complete it anytime and anywhere, avoiding disruptions to daily routines.

1.1 Research question and problems

This study aims to address critical gaps in VR-based fire evacuation training by developing a simulation tailored to Xamk's campus environment using Unity. Below are the research questions and challenges that frame this investigation.

- **What are the key technical and user experience challenges in designing an immersive VR evacuation system that balances realism with performance on accessible hardware?**
- **How does VR-based evacuation training compare to traditional methods in improving response times, evacuation times, spatial awareness, and decision-making under stress for campus occupants?**
- **How can a Unity-based VR simulation accurately replicate fire dynamics during campus evacuations to improve training realism?**

1.2 Research objectives

These questions challenged the technical and practical implications of VR-based fire evacuation training. Answering these questions especially the first question, required development of a platform for VR environment capable of simulating the Xamk campus located in Kotka. Resolving the second research question required a strategy to measure response times, evacuation times, spatial awareness, and decision-making. Finally, the study proposed using Unity packages and custom-developed systems to produce a realistic threat in this context. The steps mentioned above direct the study toward solving the following research objectives.

- **Develop a VR (VR) simulation of the Xamk Kotka campus for fire evacuation training.**
- **Address the technical and user experience challenges in creating an accessible, high-fidelity VR evacuation system.**
- **Implement and evaluate a data-driven methodology for assessing user performance and experience.**
- **Explore VR simulations' potential as a cost-effective, flexible, and less disruptive alternative to or supplement to conventional fire drills.**

The hypothesis is that using VR simulations for fire evacuations at Xamk will improve response time and evacuation time and reduce error rate without disrupting campus operations. To evaluate the hypothesis, the first step was to design and develop a 3D Unity environment replica of the Xamk campus in Kotka. This model included fire exits, rooms, furniture, and lighting to make the space feel familiar and accurate compared to a physical establishment. Secondly, the prototype included integrating visual effects to replicate fire and create the main threat for users. Thirdly, the model implements the VR component in the Unity environment through the OpenXR toolkit. Custom C# scripts were also developed, accommodating the real-world scenario and game mechanics. The last step was data collection from participants through simulation runs and surveys.

1.3 Organisation of the study

The theoretical framework section provides the theories, models, or approaches that guided this study's methodology and implementation. These frameworks guided the research design, informed the interpretation of findings, and connected the study to approaches done by other researchers. A substantial portion of this section explores previous implementations of simulation-based learning models, which informed the study's methodology and highlighted key factors such as knowledge retention. The findings also highlighted that simulation-based training was cost-effective and motivated participants, as this had been supported by most research. Additionally, this section clarifies any assumptions or perspectives shaping the study and research process, data collection, and interpretation of result.

The implementation section follows the main steps in the development cycle of the prototype. This section details the design, development, and integration of the 3D environment within the system architecture by breaking the prototype down into its main components. Similarly, this section details how visual effects, VR locomotion, data logging, and state management were designed, developed, and integrated within the system architecture. Unity 2022 LTS was the development platform, integrating Unity packages for functionality. These Unity packages included Unity VFX Graph to enhance visual effects, OpenXR to facilitate VR experiences, and Brookes et al.'s (2020) Unity Experiment Framework (UXF) to manage session creation, data logging, and result exportation. This section details the roles of 3D environment modelling, visual effects integration, VR locomotion, data logging, and state management. It explains how the strengths and limitations of these components influence the final prototype. Each element developed in this prototype section represents the transition from the initial concept to a research-ready VR fire evacuation simulation for the Xamk campus in Kotka.

The research methodology section focuses on research design, detailing participants or data sources, materials, a summary of the systems used, and the data collection procedures. This section also presents the measured variables and techniques used in the study. It goes through the qualitative and quantitative research processes, individually justifying the choice for a mixed-methodology approach. In addition, there is a description of ethical considerations for the experiments, which concerned data protection, participant consent, and participant choice to withdraw from the study.

The results section presents the study's findings using figures and tables to visualise and summarise the results meaningfully. A report on the data collected without interpretation, focusing on the raw outcomes that address the research questions or hypotheses. Additionally, this section introduces results obtained after simulation runs, including average evaluation time and differences in time between series trials and response time averages. Furthermore, this section details the process of how the raw data developed into the calculated metrics. More Particularly, this section shows how raw metrics, along with their respective algebraic equations, contributed to producing a valid evaluation and meaningful results.

The discussion section evaluates the results, explains their significance within the research context. It explores the implications of the findings, considers whether the results support the initial hypotheses, and discusses any unexpected outcomes. During this evaluation, by linking the quantitative metrics and survey responses, showed support of the findings and conclusions. Furthermore, this section mentions the study's main limitations, suggests alternative explanations, and proposes directions for future research. Along with those mentioned earlier, this segment also features suggestions and recommendations for future research, borrowing ideas from the challenges faced during the implementation and the evaluated findings. This information guides future research by suggesting areas for further exploration.

2 THEORETICAL FRAMEWORK

This section evaluates the effectiveness of VR-based training systems, analyses current technological trends, and discusses their implications for learning outcomes, focusing on fire evacuation scenarios. Drawing on published studies and the development of the Xamk campus in Kotka simulation, this analysis underscores VR's potential to improve safety training methodologies.

2.1 Environmental accuracy

A developmental factor of any simulation is how well the simulation can replicate real-life scenarios. The principle of ecological validity states that simulated environments must closely mirror real-world conditions to produce authentic human reactions (Kinateder et al. 2014). In this study, the primary environment covers the interior of the Kotka Xamk campus, which required accurate scaling and structuring to ensure ecological validity. To achieve this, the study utilized a LiDAR-scanned model produced by Toni (2024) rather than creating an FBX model from scratch in 3D modelling software. This addresses the research question about balancing technical realism with hardware performance, as even minor discrepancies in environmental layout can alter evacuation behaviour. For instance, Kinateder et al. (2014) demonstrated that participants in geometrically accurate VR environments exhibited navigation patterns 89% consistent with real-world behaviour.

Similarly, Slater et al. (2013) emphasise that ecological validity hinges on aligning sensory inputs (e.g., visual, auditory) with task demands, such as smoke obscuration or heat cues to evoke stress. Recent research highlights that even minor environmental discrepancies, like 10% errors in corridor widths, can distort evacuation decisions, reducing training efficacy (Saghafian et al. 2020). To address this, mixed-methods frameworks combine quantitative metrics (e.g., motion tracking) with qualitative feedback (e.g., Likert-scale presents surveys) to assess both behavioural and psychological fidelity (Bourhim & Cherkaoui 2020).

These papers promoted the importance of ecological validity in simulation training, which could directly affect the accuracy of research results. To ensure ecological validity, the Toni (2024) LiDAR model shall be used to implement the simulation-based training environment.

2.2 Immersive learning and HBiF models

Utilising VR for immersive learning is not an entirely new concept. Still, numerous applications employed it to create more engaging learning experiences, resulting in faster and better skill acquisition and knowledge retention. Zhu & Li (2021) present a critical review of virtual and augmented reality applications in emergency management for built environments. The research explored existing applications, examined their advantages and limitations, and considerd the future direction of these technologies in this field. The review revealed that VR/AR technologies are used for hazard recognition, prevention, and safety training. More specifically, VR/AR enables realistic, controllable simulations for evacuation and search-and-rescue research. Furthermore, experiments confirm that VR/AR can effectively replicate real-life behaviours in emergencies, though some discrepancies remain. One of the main limitations the paper lays out is that of VR-induced dizziness and nausea, with current solutions only alleviating but not eliminating these effects.

While this technique for immersive learning offered practical benefits, other training formats, such as non-interactive videos, also demonstrated options for informing potential evacuees. Chittaro & Sioni (2015) explore the implications of intractable game and non-intractable video simulations and assess the participants ' knowledge, perceived vulnerability, severity, self-efficacy, recommendation efficacy, and recommendation simplicity. The author concluded that interactive and non-intractable simulations significantly increased emergency preparedness knowledge and self-efficacy, with no significant difference. This conclusion suggests that both formats are practical for learning and confidence-building. The researchers also claim that interactivity enhances risk perception and emotional engagement, which may be necessary for motivating protective behaviour. In summary, the study demonstrates the value of grounding serious game design in psychological theory and suggests that offering both formats can maximise educational reach and impact.

Immersion in VR- the feeling of "being there"-enhances skill acquisition by triggering stress responses comparable to real emergencies (Slater et al. 2009). Studies show that high-presence VR training improves reaction times by 40%

compared to video-based drills, as the emotional engagement fosters muscle memory and decision-making under duress (Bourhim & Cherkaoui 2020). By simulating the adrenaline of a fire scenario, Bourhim & Cherkaoui (2020) test whether heightened presence translates to improved evacuation performance.

Complex interfaces can overwhelm users, reducing training effectiveness. The simulation simplifies controls and uses colour-coded path markers to minimise extraneous cognitive load, ensuring users focus on evacuation tasks rather than interface navigation. Guo et al. (2020) found that VR systems with intuitive UIs reduced user errors by 32% in high-stress scenarios, supporting the design choice to prioritise clarity over visual realism in menus. This factor informs the study's second research question, as reduced cognitive load may explain VR's potential advantage over text-heavy traditional methods.

HBiF models classify evacuation behaviours into phases: pre-movement (delayed reactions), movement (route selection), and post-movement (safety evaluation) (Gwynne et al. 2020). The simulation's data-logging system tracks these phases by recording response times, path deviations, and interaction errors (e.g., door-handling failures). For example, the arm-swing locomotion mechanic introduces variability in movement speeds (0.5–1.8 m/s), reflecting real-world diversity in mobility. This approach assesses the hypothesis that VR can replicate collective behaviours like "herding" or "freezing," addressing the research question about simulating human behaviour accurately. Gwynne et al. (2016) emphasise that behavioural realism is critical for predicting evacuation outcomes in complex buildings.

Taking the above into account, the prototype will integrate the adoption of VR for hazard recognition and evacuation training, prioritising controllable fire scenarios with adjustable difficulty. While this immersive learning technique offered practical benefits, other training formats, such as non-interactive videos, provided additional ways to inform potential evacuees.

2.3 Viability of simulation-based training

Simulation-based training (SBT) has been present for some time now, primarily used in the aviation industry for upcoming pilots to train without risk of injury. Moreover, Bilotta et al. (2013) investigated the implications of simulation-based training in the medical sector by exploring how simulation-based training can improve practitioner competency and patient safety compared to traditional training methods. The authors hypothesise that SBT, modelled after methodologies from the aviation industry, can significantly enhance learning speed, information retention, and deliberate practice and lead to safer patient care. This review article thoroughly evaluated findings from 33 papers published between 1998 and 2013. The paper highlighted the effectiveness of simulations, demonstrating that simulation-based training outperformed traditional training and apprenticeship models in skill acquisition, knowledge retention, and the safe practice of complex scenarios. Successful simulation-based training programs rely on immediate feedback, repeated practice, scenario variation, and clear learning objectives. Debriefing emerged as the most crucial element for learning and skill retention. Ruesseler et al. (2015) similarly investigated the implications of simulation-based training for medical emergencies. The researchers hypothesise that simulation-based training will lead to superior practical skills in emergency management compared to theory-based curricula. They implemented a controlled, anonymous educational trial with 44 final-year medical students at Frankfurt Medical School. During these trials, participants received a 3-day simulation-based curriculum integrating Basic Life Support (BLS), Advanced Cardiac Life Support (ACLS), and trauma management (ATLS principles), plus debriefing sessions. A control group completed the former curriculum, including lectures and emergency department shifts. Performance-based objective structured clinical examinations with 10 stations (6 scenario-based, 4 skill-based) and checklist evaluations assessed the participants. The simulation group scored 76–90% of maximum points vs. 52–62% for controls, displaying the comparison of knowledge retention. Simulation training enhanced structured decision-making, teamwork, and adherence to emergency algorithms.

Salas et al. (2009) addressed two central questions: what advantages simulation-based training offer for developing managerial competencies in educational settings, and how can educators effectively design and implement simulation-based training in management curricula to maximise learning outcomes. A seven-stage model for simulation-based training implementation, adapted from medical and aviation training, addressed these questions. More specifically, simulation-based training compressed years of on-the-job experience into controlled practice scenarios, enabling rapid skill development in decision-making, communication, and crisis management. Additionally, the author mentions the cost-effectiveness of these trainings and complements the realistic practice, which enables an engaging experience in high-stakes scenarios without any real-world consequences. The authors also noted that learners were more motivated when using interactive, game-like environments than passive methods such as lectures. In turn, this resulted in simulations outperforming traditional methods in knowledge retention and application. To achieve these results, the authors suggested collecting dynamic feedback to reinforce learning and performance metrics to assess outcomes and processes. The study outlines the key guidelines for preparing students for pre-simulation, maintaining engagement during training, and facilitating post-training skill transfer.

Bruppacher et al. (2010) investigate whether simulation-based training or an interactive seminar improves real-life patient care performance during weaning from cardiopulmonary bypass among senior anaesthesiology trainees. The authors hypothesise that simulation-based training will result in superior acquisition and transfer of technical and nontechnical skills to the clinical environment compared to traditional seminar-based teaching. A simulation-based training group received a 2-hour individual session using high-fidelity simulation, including four crisis scenarios, a realistic operating room setup, and structured debriefing. Another interactive seminar group received a 2-hour individual seminar with audiovisual aids and paper-based scenarios, covering the exact content domains. Both groups improved from pre-test to post-test and retention test, but the simulation group showed significantly greater improvement. Simulation-based training led to superior technical and nontechnical skills transfer into real-life clinical performance during CPB weaning compared to interactive seminars. The immersive, realistic, and feedback-rich environment of simulation-based training contributed to these gains, particularly in nontechnical skills such as communication, leadership, and decision-making. Results support integrating high-fidelity SBT into residency programs for crisis management and complex procedures, potentially reducing patient risk during trainee learning.

3 IMPLEMENTATION

This section breaks down the development and deployment of the final prototype into its components, with particular emphasis on the technical processes and decisions that maximise ecological validity. Having leveraged Unity 2022 LTS and techniques to simplify modelling, the implementation aimed to replicate the real-world campus environment with high fidelity, ensuring that users' virtual space experiences closely mirror those in actual emergency scenarios.

Aspects of the implementation included the construction of a LiDAR-based 3D campus model, integrating dynamic fire and smoke effects using Unity's VFX Graph, and developing custom VR locomotion systems to reflect authentic human movement. The section also explores the adoption of the Unity Experiment Framework (UXF) for data collection and using singleton classes for robust state management and cross-scene data persistence. By breaking down each technical element, ranging from environmental modelling and visual effects to interaction design and data logging, this section demonstrates how the prototype was engineered to address the research questions, support empirical testing, and provide a scalable foundation for future safety training applications.

3.1 Map creation and model development

Accurate spatial representation is essential for ecological validity and ensured that user navigation, decision-making, and stress responses in VR closely mirror real-world behaviour (Kinateder et al., 2014; Dakeev et al., 2020). An ecological valid model reduces the risk of users learning unrealistic or unsafe evacuation habits. Firstly, a 3D environment development, aimed to be a space that participants could match with the real-world equivalent. Xamk game studios provided a LiDAR FBX model created by Toni (2020), which established itself as an accurate model of the real-world campus and reduced development time. Using LiDAR-scanned FBX models improved the realism of the simulations by accurately reproducing the campus layout's scale, position, and orientation.

The effectiveness of VR research tools depended on the accuracy of the model when compared to the real world, as a lack of ecological validity would have posed a significant risk to their utility (Kinateder et al., 2014, pp. 313-321). This approach aligned with VR training methodologies prioritising environmental validity, showing how precise geometry affected user navigation behaviour and stress response accuracy (Dakeev et al., 2020; Yu, 2022). For example, research by Kinateder et al. (2014) highlighted that even minor deviations, such as 10% differences in corridor widths, could have affected people's decisions during evacuations.

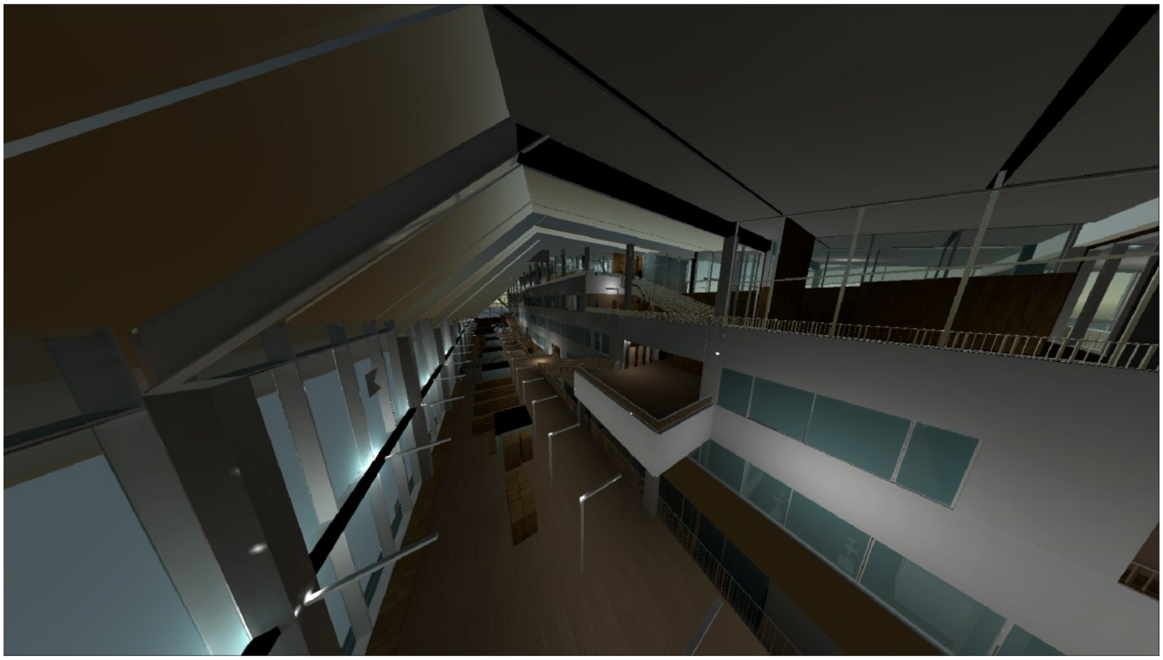


Figure 1. The finalized LiDAR Xamk campus

Figure 1 displays the interior of the virtual XAMK campus, after having filtered the LOD of the model to the most accurate one and adding baked lighting to the scene for further realism and variability to the meshes. The campus comprises of approximately 7100 individual meshes, all with the materials and mesh collider components for gameplay purposes. The lighting comprised pointing lights to simulate the indoor environment but also included a singular directional light to replicate the sun. Furthermore, to reduce performance issues, light baking was used, which resulted in higher frame rates and less screen tearing.

In its initial format, the model developed by Toni (2020) and provided by Xamk Game Studios contained multiple virtual repetitions of the Xamk campus. During an investigation using 3D modelling software, models labelled with the postfix "LOD0”, had the best level of detail. Therefore, the next step was to eliminate "LOD1" to "LOD3" model renditions, which proved challenging, as this would usually crash the 3D modelling software. Additionally, importing "LOD0" models as a whole structure would create a static model asset that did not allow editing of specific sub-models, had a lengthy loading time in Unity, and adding or editing mesh colliders could only be done collectively.

The strategy to overcome both challenges was batch exportation using a Python script in the 3D modelling software. While it was a prolonged process, this strategy resulted in the exportation of 7100 individual FBX models in the Unity project, hence this separation ensued in the editing of individual models to fit the further target environment.

Adding lighting made the map's visuals more dynamic and realistic, but system performance slowed down during initial testing. The main issue with runtime lighting is that it requires excessive calculations per update. The solution was using baked lighting, a more static but extremely efficient lighting pipeline. This pipeline calculates the lighting beforehand and saves it to an image file known as a lightmap. While quality would make lighting in the model look more pleasing, it was not within the scope of this study. Hence, the lightmap parameters chosen would be normal to inferior quality, having a lightmap resolution of thirty texels per unit. Initially, the resulting lighting took a long time to generate, but barely affected performance during the simulations, as runtime had no lighting calculations.

3.2 Visual effects for simulations

The presence of realistic hazards is necessary for simulating urgency and influencing user behaviour during evacuation (Bourhim & Cherkaoui, 2020; Guo et al., 2020). Visual effects help participants recognize danger zones and make more authentic decisions under stress. The initial system used was a default particle system using an edited flare shader. The dilemma with the initial effects was that they were not sufficiently close to a real fire, which severely broke immersion. The main aim was for the digital environment to incorporate practical additions that better fit the situation, but these effects had to represent real-world threats closely. Therefore, the Unity VFX package was used instead, which enabled the development of visually realistic particle effects on meshes. The resulting system, as presented in Figure 2, included both flame and smoke, which further amplified the reliability of the prototype. This component was a considerable strength of VR experiments.

Additionally, with integrated audio cues in both the main menu and simulation scenes, situational awareness was not dependent solely on visuals. The main menu introduced the user to a calming melody to produce a sense of safety, while the main game scene invoked stress with a fire alarm sound. The reasoning behind this was to reflect the stillness of an uneventful day at an institution and then transition into a more panicked environment.

Figure 2. VFX Fire Output example

To produce the fire VFX as displayed in Figure 2, the Figure 3 VFX graph had to be designed. The initially developed effect was a static visual component, but testing on different models revealed that particles on larger models lacked sufficient density. Therefore, to create a more controllable and suitable particle effect, new parameters for density and particle spread were updated to be tuneable, ensuring generated particles were consistent on all meshes. The main parameters used were Smoke Rate (float), Flame Rate (float), Transform, and Mesh, which, by editing their values in the Unity inspector, allowed fire effects to match the environment more clearly. Two central particle systems were set up, one for smoke and one for flames, which both had a “Rate” parameter, which, when increased or decreased, affected spawn rate and density. Moreover, randomly generated particles would be generated on a mesh surface of an ignited object through a reference of a mesh and transform. By balancing these parameters, the results-maintained performance efficiency across various platforms and could be applied to all meshes on the map.

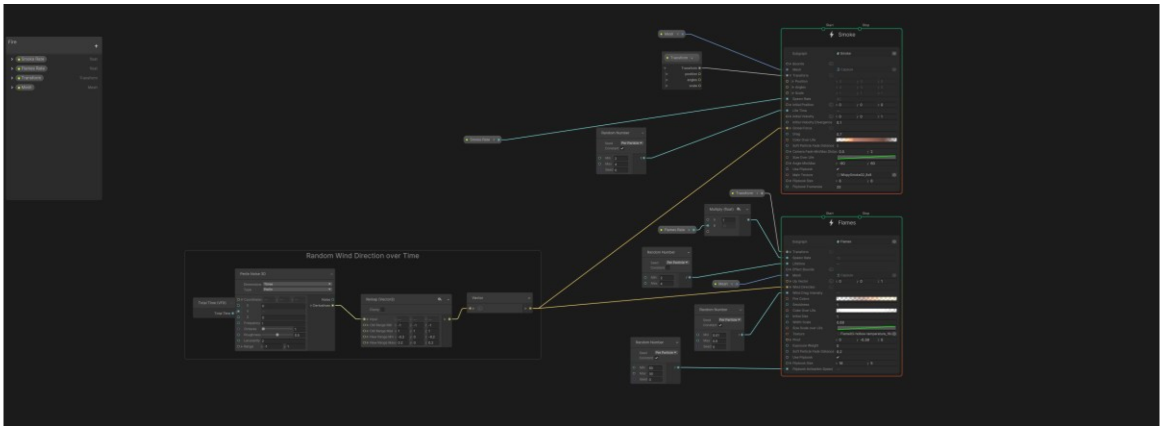


Figure 3. Fire VFX Graph

3.3 Virtual reality interaction and movement functionality

Unity offers numerous ways to integrate VR into the Unity engine. The initial integration used the Pico Unity Integration SDK, and although it benefited from reduced hardware requirements, issues arose that could have severely slowed down development. This pipeline functioned on Android-based VR devices, which could have limited the tool’s usability to specific and circumstantial hardware. Furthermore, the PICO SDK locomotion system was limited to teleportation, which did not match the requirements of the study. In contrast, the OpenXR provided a standardised interface and better support for hand-tracking and custom movement scripts directly documented by Unity.

Therefore, the final prototype used the OpenXR toolkit, which provided a standardised interface and enhanced support for hand-tracking and custom movement scripts directly documented by Unity. The simulation can function seamlessly on a Windows-based computing unit, which is connected to a configured virtual reality (VR) headset for input and output. Moreover, the development of a custom locomotion script took place, so the evacuees’ speed depended on how fast they swung their arms. Other VR locomotions (e.g., joystick, teleportation) did not reflect human movement and could have reduced realism. By tying movement speed to arm-swing velocity, the simulation more accurately mimicked physical actions and variability among participants, which was crucial for studying evacuation dynamics and fatigue (Slater et al., 2009). In general, movement speed became proportional to physical arm swings.

This approach also allowed for finer control over acceleration and deceleration, making navigation through narrow corridors and around obstacles more natural and intuitive.

Figure 4 displays an example of the user experience through the OpenXR origin prefab. Customising this prefab followed in interactions that fitted the situation and answered the research questions more clearly. Additionally, changing the default OpenXR controllers to the OpenXR hands further enhanced immersion.

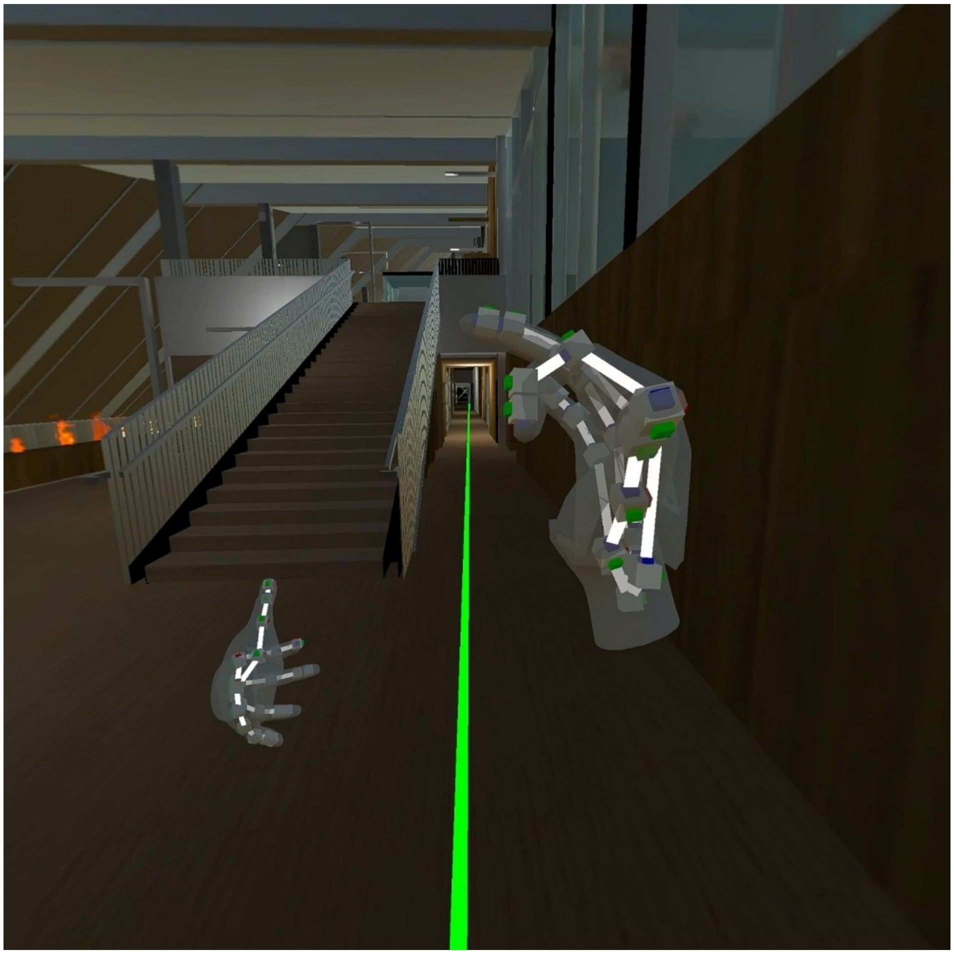


Figure 4. Classroom assisted simulation example.

While a straightforward locomotion system, which operated through hand tracking to add velocity to the Player game object, was developed, an issue arose with rigid movement. The resulting movement experience was very unnatural and unrealistic. Therefore, the final arm-swing locomotion system shown in Figure 5 employed a tuneable FIFO buffer to smooth hand velocity data, addressing tracking inaccuracies inherited in VR hardware. The system filtered high-frequency noise while preserving intentional motion signals. The size of the buffer was adjustable to fetch the best configuration and movement. Larger buffers (e.g., 10 frames) increased stability but introduced 120–150ms latency, potentially breaking immersion and low buffers size would create irregular movement. The default 5-frame balance aligned with VR locomotion best practices. Movement activation followed a linear response curve, where velocity (v) maps to variable speed (s) through a tuneable function:

$$s = \omega \cdot R(v)$$

Equation 1. Game Manager velocity equation.

Here, ω represented the maximum walk speed (5 m/s), and R(v) is the evaluation of the speedResponseCurve. Moreover, adding vertical physics at a rate of 9.8m/s with the _gravity vector to the horizontal arm-swing input vector allowed the OpenXR origin to move in-game realistically. Additionally, multiplying time between frames with the pre-mentioned sum of vectors resulted in physics being independent of the frame rate. For instance, users running at maximum speed on a system running 30 FPS would cover the same distance as if running at the same speed on a 120FPS system. The 120FPS system provides a frame interval one quarter that of 30FPS, prompting a resultant scaling of the vector time.

```
public class WalkInPlaceLocomotion : MonoBehaviour
{
    [Header("References")]
    [SerializeField] private CharacterController characterController; // Reference to the CharacterController for movement
    [SerializeField] private GameObject leftHand;                    // Reference to the player's left hand GameObject
    [SerializeField] private GameObject rightHand;                   // Reference to the player's right hand GameObject

    [Header("Settings")]
    [SerializeField, Range(0f, 100f)] private float walkSpeed = 5f;  // Maximum walking speed
    [SerializeField, Range(0, 10)] private int velocityBufferSize = 5; // Number of frames to average for velocity smoothing
    [SerializeField] private AnimationCurve speedResponseCurve = AnimationCurve.Linear(0, 0, 1, 1); // Curve to map hand velocity to movement speed

    // Internal state
    private bool moved = false;                                      // Tracks if movement occurred (used for response time update)
    private Vector3 _previousPosLeft, _previousPosRight;             // Previous positions of left and right hands
    private Vector3 _direction = Vector3.zero;                       // Smoothed movement direction
    private Vector3 _gravity = new Vector3(0f, -9.81f, 0f);          // Gravity vector

    private Queue<float> _velocityBuffer = new Queue<float>();       // Buffer to smooth hand velocity
    private float _smoothedVelocity = 0f;                            // Smoothed velocity value

    /// <summary>
    /// Called when the script instance is being loaded.
    /// Checks for required components and falls back if missing.
    /// </summary>
    Unity Message | 0 references
    private void Awake()
    {
        if (!CheckComponents())
        {
            Debug.LogError("Missing VR components! Falling back to default locomotion.");
            var fallback = GetComponent<ActionBasedContinuousMoveProvider>();
            if (fallback != null) fallback.enabled = true; // Enable fallback locomotion if available
            enabled = false; // Disable this script
        }
    }

    /// <summary>
    /// Checks if all required components are assigned.
    /// </summary>
    1 reference
    private bool CheckComponents()
    {
        return characterController != null
            && leftHand != null
            && rightHand != null
            && Camera.main != null;
    }

    /// <summary>
    /// Initializes previous hand positions at the start.
    /// </summary>
    Unity Message | 0 references
    private void Start()
    {
        _previousPosLeft = leftHand.transform.position;
        _previousPosRight = rightHand.transform.position;
    }

    /// <summary>
    /// Updates hand velocities and smooths them using a buffer each frame.
    /// </summary>
    Unity Message | 0 references
    private void Update()
    {
        // Calculate current frame hand velocities
        Vector3 leftHandVelocity = leftHand.transform.position - _previousPosLeft;
        Vector3 rightHandVelocity = rightHand.transform.position - _previousPosRight;

        // Combine velocities (weighted sum)
        float totalVelocity = leftHandVelocity.magnitude * 0.8f + rightHandVelocity.magnitude * 0.8f;

        // Add to velocity buffer for smoothing
        _velocityBuffer.Enqueue(totalVelocity);
        if (_velocityBuffer.Count > velocityBufferSize)
            _velocityBuffer.Dequeue();

        // Compute smoothed velocity (average of buffer)
        float sum = 0f;
        foreach (float v in _velocityBuffer) sum += v;
        _smoothedVelocity = sum / _velocityBuffer.Count;

        // Store current positions for next frame
        SetPreviousPos();
    }

    /// <summary>
    /// Moves the character based on smoothed velocity and direction in FixedUpdate for physics consistency.
    /// </summary>
    Unity Message | 0 references
    private void FixedUpdate()
    {
        Vector3 movement = Vector3.zero;
        Vector3 combinedMovement;

        // If the user is moving their hands enough, move forward
        if (_smoothedVelocity > 0.01f)
        {
            // Project camera's forward onto the horizontal plane to get movement direction
            Vector3 projectedForward = Vector3.ProjectOnPlane(Camera.main.transform.forward, Vector3.up).normalized;

            // Smoothly interpolate the movement direction
            _direction = Vector3.Lerp(_direction, projectedForward, 5f * Time.fixedDeltaTime);

            // Use animation curve to map velocity to speed response
            float response = speedResponseCurve.Evaluate(Mathf.Clamp01(_smoothedVelocity));
            movement = _direction * (walkSpeed * response * Time.fixedDeltaTime);
            Debug.Log($"Smoothed Velocity: {_smoothedVelocity}");

            // Combine movement with gravity
            combinedMovement = movement + _gravity * Time.fixedDeltaTime;

            // Optional: Update response time if movement occurred
            if (moved!)
            {
                GameManager.Instance.updateResponseTime();
                moved = true;
            }
        }
        else
        {
            // If not moving, only apply gravity
            combinedMovement = Vector3.zero + _gravity * Time.fixedDeltaTime;
        }

        Debug.Log($"Movement: {combinedMovement}");

        // Move the character controller
        characterController.Move(combinedMovement);
    }

    /// <summary>
    /// Updates the stored previous hand positions for velocity calculation.
    /// </summary>
    1 reference
    private void SetPreviousPos()
    {
        _previousPosLeft = leftHand.transform.position;
        _previousPosRight = rightHand.transform.position;
    }
}
```

Figure 5. Locomotion custom code for OpenXR movement

3.4 Management and Singleton development

The prototype's architecture required a structure to manage dynamic state transitions and store global variables across scenes. Components like UI, VR locomotion, and experimental sessions (UXF) required access to this shared data. To fulfil these requirements, the assessment of multiple systems led to the decision to utilise classes implementing the singleton pattern.

The Singleton pattern ensured that only one instance of each class and its properties existed during runtime, providing a global point of access for session management and data storage. Moreover, between runs, properties would be reset to default, allowing each session to start unassisted, which resulted in participants' simulation initiation with minimal updates to the simulation settings. Data storage systems, such as Unity Player Preferences, would carry over data between runs, which is why singletons were the best direction for development.

The centralised structure allowed for faster troubleshooting, as most custom properties and functionalities were in one place, enabling the identification of bugs using breakpoints. Other considered approaches, such as passing data via scene parameters or using static classes, were unsuitable due to Unity's scene management lifecycle and the need for encapsulation. The design for these classes ensured adherence to standard code practices, such as encapsulation and reduced redundancy.

The Game Manager class functioned as the state management hub, managing workflows such as scene transitions, UXF session initialisation, and simulation timing. For example, its “LoadMainScenario()” method dynamically configured spawn points based on participant routes while integrating with Unity's SceneManager and “DontDestroyOnLoad” to preserve continuity between menus and VR environments. Table 1 explains the methods and coroutines for this class.

Method Name	Description
Awake	Initializes the singleton instance, ensures only one exists, and subscribes to scene change events.
Generate	Sets up experimental blocks and catch trials for a new UXF session.
setMode	Sets the current simulation mode (e.g., Assisted, Fire) via GameData.
setRoute	Sets the player's spawn point based on the selected route.

OnDestroy	Unsubscribes from scene change events on destruction of the object.
OnDisable	Unsubscribes from scene change events on disabling of the object.
OnActiveSceneChanged	Manages logic for loading scenarios or menus when the active scene changes.
ChangeScene	Initiates a coroutine to load a new scene asynchronously.
SceneChanging (Coroutine)	Loads a new scene asynchronously using Unity's SceneManager.
LoadMainSenario	Configures the simulation environment, sets player position, and starts the simulation.
StartSimulation (Coroutine)	Waits for a random delay, then enables audio, VFX, and player locomotion, and logs the start time.
LoadGameMenu	Activates and configures the main menu UI for user interaction.
FinishSimulation	Logs end time and duration, ends the current trial, and returns to the main menu or exits the app.
MakeCatchTrials	Modifies selected trials to serve as catch trials and shuffles their order within a block.
updateResponseTime	Updates and logs the participant's response time for the current trial.

Table 1. Game Manager class methods and coroutines

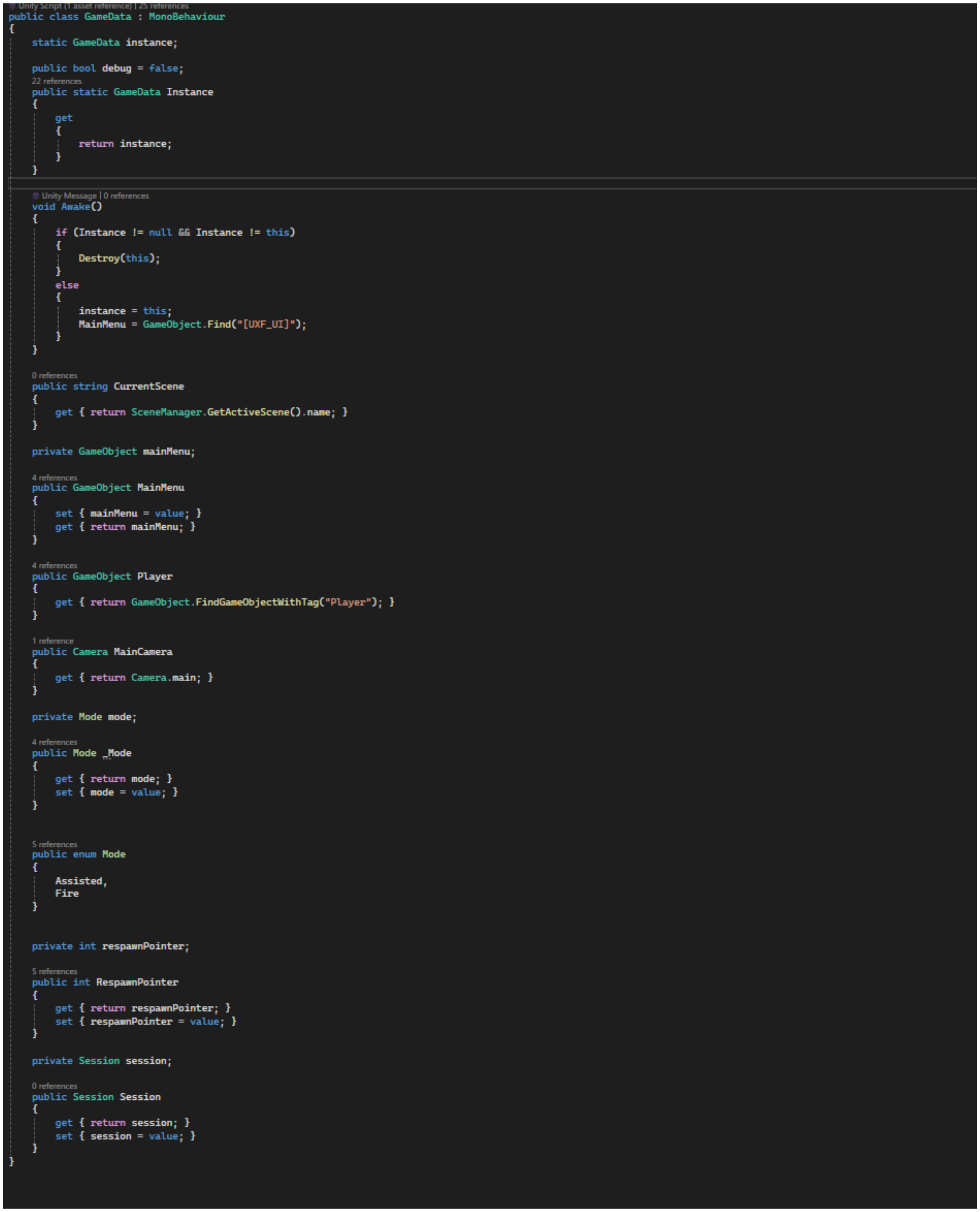
The GameData class was a centralised, singleton-managed repository for game state and configuration data within the Unity-based VR fire evacuation simulation. By implementing the singleton pattern, GameData had ensured that only one instance persisted across all scenes, maintaining access to global variables such as the current simulation mode, player reference, camera, session data, and UI elements. The class exposed multiple properties, including Main Menu for the main UXF-driven user interface, player for access to the player Game Object (using

Unity's tagging system), and Main Camera for referencing the scene's primary camera, user interactions, and immersive VR experiences.

The Current Scene property utilised Unity's SceneManager to provide the active scene's name, which supported context-sensitive logic such as environment initialisation or UI updates. A key feature of the class is the Mode enumeration, which distinguished between "Assisted" and "Fire" simulation trials. The "Assisted" mode offered a guided experience with visual cues, while the "Fire" mode provided an unassisted environment to evaluate participants' fire evacuation skills. This mode was set and retrieved via the “_Mode” property and was essential to both the game's logic and the data collected through the UXF (Brookes et al., 2020). The Respawn Pointer integer tracked the player's spawn location and forwarded scenario setups and route selection settings. The Session property was linked to the current UXF session and enabled control over the experiment and data logging.

The architecture enhanced maintainability and scalability by splitting data storage (Game Data) from methodical logic (Game Manager). This separation allowed for communication between core systems and reduced over dependency on a singular script. For instance, Game Manager could have updated the simulation mode or spawn point by modifying GameData properties, which influenced the initialisation of scenes, the configuration of trials, and participant data analysis. The class’s design also integrated with Unity’s Input and scene management systems, maintaining accurate player and camera references even as scenes changes or prefabs swaps.

These classes were the centre point of the simulation-based training model, as they ensured that all system implemented functioned simultaneously, and mainly connected gameplay to its core components. For instance, depending on UI choices the "Changescene()" method would position the Player to one of the starting point game objects accordingly. This approach enabled the model to remain scalable and dynamic. The game manager also managed results exported during testing, ensuring the validity and accuracy of all collected data.



```
Unity Script (1 asset reference) | 25 references
public class GameData : MonoBehaviour
{
    static GameData instance;

    public bool debug = false;
    22 references
    public static GameData Instance
    {
        get
        {
            return instance;
        }
    }

    Unity Message | 0 references
    void Awake()
    {
        if (Instance != null && Instance != this)
        {
            Destroy(this);
        }
        else
        {
            instance = this;
            MainMenu = GameObject.Find("[UXF_UI]");
        }
    }

    0 references
    public string CurrentScene
    {
        get { return SceneManager.GetActiveScene().name; }
    }

    private GameObject mainMenu;

    4 references
    public GameObject MainMenu
    {
        set { mainMenu = value; }
        get { return mainMenu; }
    }

    4 references
    public GameObject Player
    {
        get { return GameObject.FindGameObjectWithTag("Player"); }
    }

    1 reference
    public Camera MainCamera
    {
        get { return Camera.main; }
    }

    private Mode mode;

    4 references
    public Mode _Mode
    {
        get { return mode; }
        set { mode = value; }
    }


    5 references
    public enum Mode
    {
        Assisted,
        Fire
    }


    private int respawnPointer;

    5 references
    public int RespawnPointer
    {
        get { return respawnPointer; }
        set { respawnPointer = value; }
    }

    private Session session;

    0 references
    public Session Session
    {
        get { return session; }
        set { session = value; }
    }
}
```

Figure 6. GameData Singleton code snippet

3.5 Data logging and user interface

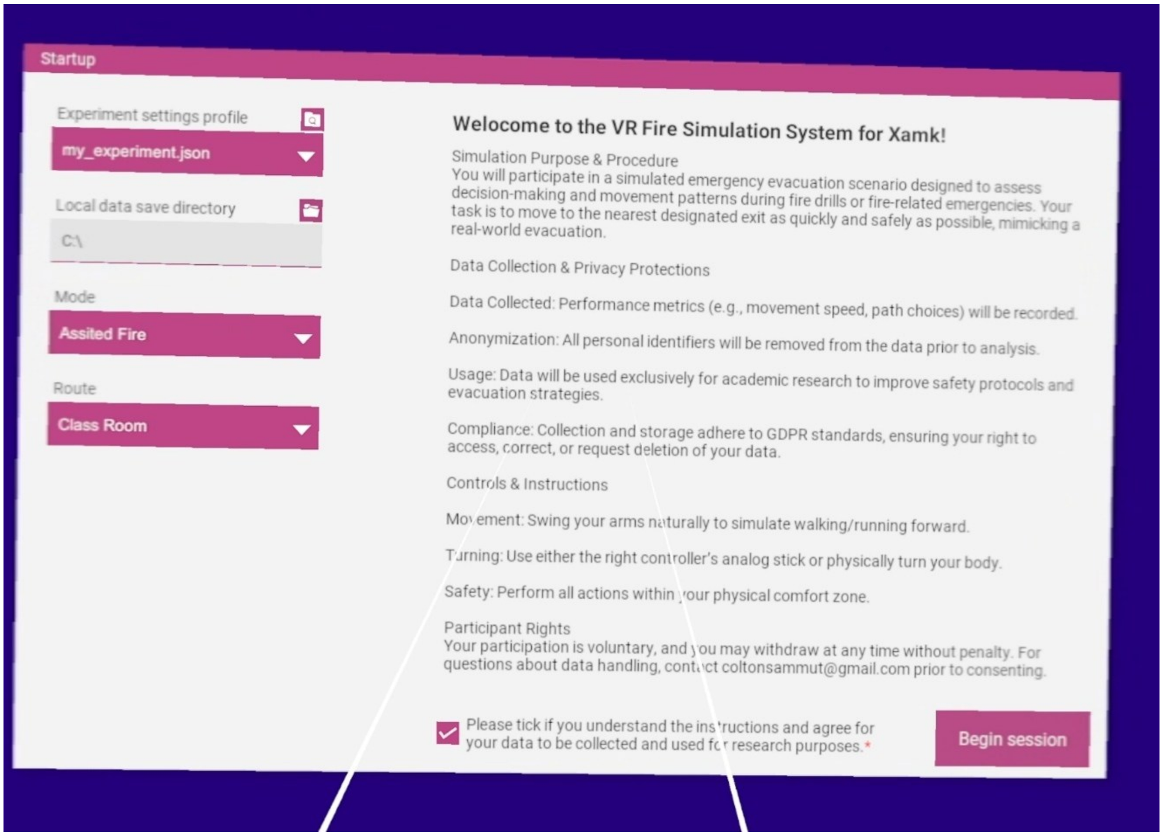
Robust data collection is necessary to evaluate user performance, response times, evacuation duration, and key outcomes. Moreover, the UXF validated the effectiveness of the simulation and supporting the research questions (Brookes et al., 2020). This package along with runtime calculations enabled the exportation of key metrics such as mode, route, participant ID, trial number, start time, end time, evacuation duration, and response metrics. This system functioned by creating a session on start-up of the first trial, and a block of empty trials. When finishing a trial, the state management would update the results of the current trail and save it to the block. After completing all trials, the system terminated the session and saves the data as a CSV file in a specified location on the local drive.

This study focused on two primary data points: response time and evacuation duration. Institution fire management (e.g., Fire wardens) usually record these metrics and gather them collectively. Hence, developing or integrating a logging system enabled replication of this data recording. While developing a custom system was viable, Brookes et al. (2020) Unity Experiment Framework (UXF) was the solution for the final prototype as it saved time and proved its robustness in its logging capabilities. The main challenge of integrating software developed by others is synchronising the integrated system to the developed one to reduce exceptions and inaccuracies and understand the integration of system components. Using Brookes et al. (2020) documentation and some testing, the UXF system was set up to record and set game settings.

Before interacting with the UXF, the system required the definition of data points and settings. The Session class resided on the "[UXF_Rig]" prefab, which served as the centrepiece of the UXF. Accessing the "Session" class through the Unity Inspector allowed for changing behaviour, data collection, data handling, and events. The main additions in the data collection section included the custom headers labels "Start Time," "End Time," "Duration," and "Response Time", which ensured that each time a trial ensues, these details appear in the records. All header data would be updated by accessing the Session class singleton instance and using the "currentTrial.results[‘Header name’]" method. The Game Manager class called classes at different simulation points to record time in seconds and get accurate time stamps. Another behaviour set up was enabling the "Dont Destroy on Load New Scene" Boolean value, which allowed the Session singleton class to keep itself in the main thread even when switching scenes. Initial testing identified the need to save the settings route and mode for evaluation, since these factors required consideration.

Similarly to the "Custom Headers," the prototype had the addition of "Mode" and "Route" data points to the "Setting To Log" header list. Instead of updating values with "currentTrial.results[“Header name’]", they used "currentTrial.settings[‘Header name’]". Finally, subscribing the "Generate Method" to the "On Session Begin" event allowed the instantiation of experimental trail blocks and catch trials for a new UXF session. While the session and trial ran, the Game Manager’s instance populated, updated, and retrieved all current header data. For instance, by pulling the mode, the game manager would determine if pathing should be enabled or not. The UXF significantly reduced development time and enabled integration of a modular, expansive system into the final prototype. Moreover, rather than recording times collectively, this system collects times of individuals showing an advantage of automatic data collection. Since the process operated automatically, only one individual was needed to participate in the evacuation drill. Additionally, recording data over multiple trials enabled rapid evacuation resets within seconds.

The UXF package also provided the main UI, as shown in Figure 7. Through the UI participants could set the session parameters such as choosing the mode (assisted or unassisted), changing the route (Classroom, Roof, Library or Hallway), and changing the local save directory. Moreover, the system logged both the mode and the route in the CSV file to provide clarity during evaluation. Furthermore, the package generated participant aliases, which the evaluation used to refer to participants and maintain anonymization. The UI communicated with state management scripts and updated the GameData class’s properties, ensuring synchronization between logged data and simulation parameters. Additionally, the UI informed the participants with their agreement of their inclusion in the study as well as controls of the simulations. Participants who did not tick the box and therefore did not agree could not participate in the experiments, as allowing them would violate GDPR.



Experiment settings profile: my_experiment.json

Local data save directory: C:\

Mode: Assited Fire

Route: Class Room

Welcome to the VR Fire Simulation System for Xamk!

Simulation Purpose & Procedure
You will participate in a simulated emergency evacuation scenario designed to assess decision-making and movement patterns during fire drills or fire-related emergencies. Your task is to move to the nearest designated exit as quickly and safely as possible, mimicking a real-world evacuation.

Data Collection & Privacy Protections

Data Collected: Performance metrics (e.g., movement speed, path choices) will be recorded.

Anonymization: All personal identifiers will be removed from the data prior to analysis.

Usage: Data will be used exclusively for academic research to improve safety protocols and evacuation strategies.

Compliance: Collection and storage adhere to GDPR standards, ensuring your right to access, correct, or request deletion of your data.

Controls & Instructions

Movement: Swing your arms naturally to simulate walking/running forward.

Turning: Use either the right controller's analog stick or physically turn your body.

Safety: Perform all actions within your physical comfort zone.

Participant Rights
Your participation is voluntary, and you may withdraw at any time without penalty. For questions about data handling, contact coltonsammut@gmail.com prior to consenting.

☑ Please tick if you understand the instructions and agree for your data to be collected and used for research purposes.*

Figure 7. UXF GUI for VR simulations

4 RESEARCH METHODOLOGY

The methodology integrated 3D environment design, user interaction modelling, and data-driven analysis to evaluate the hypothesis that VR simulations improved evacuation response times and evacuation time. Moreover, the methodology incorporated a survey and observations to gain insight into the user experience and their knowledge of the topic. Combining quantitative metrics with qualitative surveys aligned with the best practices in evaluating VR-based research (Bugli et al., 2023). This triangulation ensured behavioural outcomes and an accurate user experience evaluation; hence, this would have answered the question regarding simulation efficacy.

4.1 Quantitative data collection

This phase focuses on the participants' simulations and the logging of data through the UXF (Brookes et al., 2020) package, such as participant ID, trial number, response time, and duration. This allowed for further analysis and interpretation of the saved data located in a CSV file without human intervention.

The three-stage simulation (two assisted, one unassisted) follows VR trial design principles from clinical VR research, balancing skill acquisition with competency testing (Birckhead et al., 2019). Assisted would have displayed the safest path the participant should take to evacuate the premises. The Third simulation would have participants doing the same evacuation, but the path would be invisible. This unassisted trial assessed the participants' knowledge of the path and exit point.

The main hardware used was the Pico 4 VR headset, Pico 4 VR hand controller, and a desktop setup. The simulation's build was an EXE, meaning it would have run on a desktop PC. The participant would have been able to interact with the environment through the PICO Connect app and Steam VR. While this approach may seem restrictive, using a PC over the VR allowed the application to run on different VR varieties rather than just standalone.

The experiments took place in Classroom 4019 in the Xamk Campus located in Kotka. Therefore, the chosen route for all simulations was the classroom route. This route would start from classroom 4019 and end at exit point 3. Furthermore, the study justified the chosen route by noting that Game Development Degree students formed the main demographic, making it the most practical option. This decision also provided participants with a sense of familiarity during the simulations (Dakeev et al., 2020). Participants started in the classroom 4019 location, chosen based on spatial familiarity studies showing 31% faster route recall in known environments (Dakeev et al. 2020). The total experiment time was between five to ten minutes, which included briefing and an explanation of the project to participants, conduction of assisted and unassisted simulation runs, debriefing, and forwarding the survey.

4.2 Qualitative data collection

During the simulations, the participants noted observations about the effects of the simulations on participants. Furthermore, following the simulations, participants filled in a survey using the Likert scale and open-ended statements to gain the participants' perspective and identify the strengths and weaknesses, opportunities, and threats that the VR simulations presented. This data collection allowed this study to gauge the simulations' realism and participant decision-making by constructing a SWOT analysis.

The Likert scale survey designed for this study served as a tool to evaluate participants' experiences and perceptions following their engagement with the VR fire evacuation simulation. Comprising ten statements, the survey employed a standard five-point Likert scale, ranging from "Strongly Disagree" to "Strongly Agree." This structure enabled participants to express the degree to which they agree with each statement and provided insights into the effectiveness and realism of the VR training experience.

The first two questions focused on the sense of presence within the simulation. The questions questioned whether the VR scenario made the participants feel as if they were experiencing a real fire emergency and whether it prompted genuine, emergency-like responses. These items were crucial for assessing the immersive quality of the simulation and its ability to elicit realistic reactions, which are key indicators of practical VR-based training.

Questions three and four shift the focus to situational awareness. By asking about participants' confidence in their ability to evacuate safely and their understanding of exits and obstacles compared to traditional drills, the survey gauged the educational impact of the VR experience. These responses assisted in determining whether VR training translates into practical preparedness and improved spatial understanding.

The fifth statement addressed VR's potential as a training modality, inviting participants to consider whether VR-based drills could replace or supplement conventional fire drills. This question directly informed the study's broader objective of evaluating VR's viability in institutional safety training programs.

Questions six and seven assessed the VR system's usability and advantages. Participants reflected on the navigation using the provided controls and whether the VR drill was less disruptive to their daily routines than traditional drills. These aspects are important for understanding VR training solutions' practical feasibility and user acceptance.

The final three questions explore fidelity, clarity, and advocacy related to the VR experience. Participants evaluate the realism of the virtual campus, the clarity of instructions and feedback, and their willingness to recommend VR training to others. These items collectively provide a holistic view of the simulation’s strengths and areas for improvement.

4.3 Ethical considerations

Developing and deploying VR simulations for emergency training required ethical considerations, particularly when human participants were engaging with environments designed to replicate high-stress scenarios. At the core of this study lies a commitment to safeguarding participant welfare, ensuring transparency in data practices, and upholding the integrity of the research process.

Informed consent formed the fundamental pillar of ethical engagement. Before participation, individuals received documentation outlining the simulation's purpose, potential risks (including mild cybersickness or temporary disorientation), and their right to withdraw without penalty. Crucially, the consent protocol explicitly clarified the artificial nature of the fire emergency, reducing undue psychological harm while preserving the scenario's training efficacy.

Data protection adhered to privacy-by-design principles throughout the project lifecycle. All datasets were anonymised at the capture point using the Unity Experiment Framework's automated aliasing system. Integrating Google Forms for post-simulation surveys intentionally excluded email collection, ensuring respondent anonymity while gathering qualitative feedback. These complied with the EU GDPR.

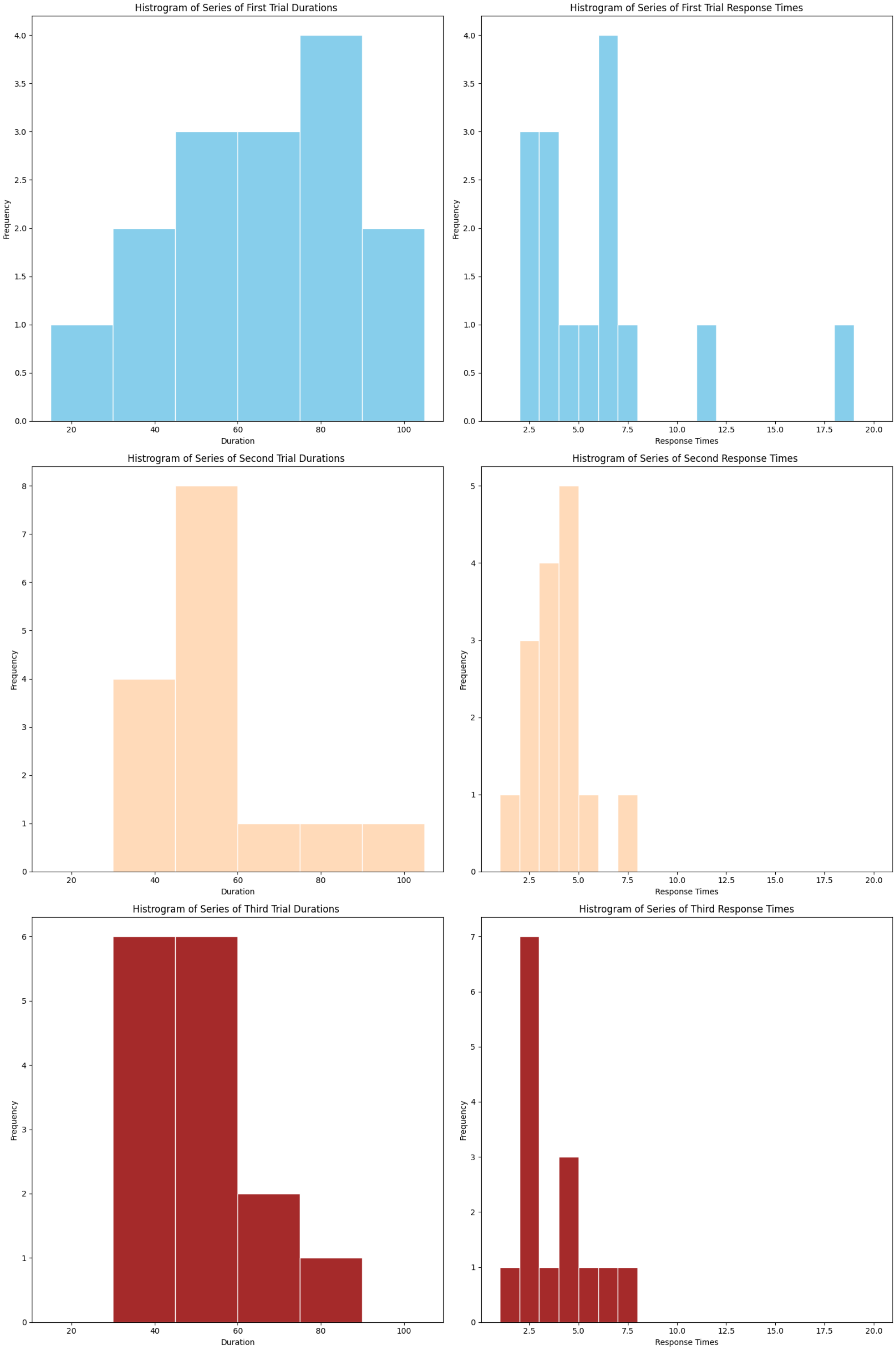
5 RESULTS AND FINDINGS

This section explores the process of recording and compiling each series of trials. More specifically, it delves into the frequency distributions to visualize patterns and variations, providing clearer insights for data interpretation.

5.1 Quantitative results

Figure 8 presents the quantitative data collected from the VR fire evacuation simulations. This figure consists of six histograms arranged in three rows and two columns. The left column displays the distribution of evacuation durations (in seconds) across the three sequential trials done during data collection, while the right column illustrates the respective response times (in seconds) for the same trials. The first step of evaluation used Python’s Matplotlib library to process data from the Unity Experiment Framework (UXF) logging system, enabling the creation of histograms with their corresponding data. Moreover, colour-coding (blue for the first, orange for the second, and red for the third trials) further simplifies the association with the respective trial series. All horizontal axes represent a range of time measured in seconds, while the vertical axes show frequency counts of participants achieving time values. This representation effectively illustrates the distribution and shows the frequency of all the metrics and their patterns in an organised and concise manner, for the 45 trials.

Figure 8's histogram collectively classifies all 135 evacuation durations and 135 responses in their respective bins. The histograms classified the bins for evacuation durations in 15-second intervals, covering a total range from 15 seconds to 105 seconds. The bins' range for response time starts from 1 second to 20 seconds at a 1-second interval. While only the first response time histogram classifies beyond 8 seconds, this range was still set for all histograms for clearer data analysis.



Histrogram of Series of First Trial Durations

Histrogram of Series of First Trial Response Times

Histrogram of Series of Second Trial Durations

Histrogram of Series of Second Response Times

Histrogram of Series of Third Trial Durations

Histrogram of Series of Third Response Times

Figure 8. Histograms of a series of trial metrics

Calculated Metric Name	Calculated Metric Figure (2 d.p)	Underflow/Overflow percentage from previous trails (0 d.p)
First trial evacuation durations average	65.87s	N/A
Second trial evacuation durations average	51.88s	- 21%
Third trial evacuation durations average	50.40s	- 3%

Table 2. Trial Evacuation Durations Averages comparison table.

Calculated Metric Name	Calculated Metric Figure (2 d.p)	Underflow/Overflow percentage from previous trails (0 d.p)
First trial evacuation response times average	6.08s	N/A
Second trial evacuation response times average	3.81s	- 37%
Third trial evacuation response times average	3.67s	- 4%

Table 3. Trial evacuation durations means comparison table.

Table 2 and Table 3 display the calculated evacuation durations, response time averages, and underflow as percentages of all the trials, respectively.

$$\bar{A} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad where\ x_i \geq 0,\ n > 0$$

Equation 2. Average for series of metrics equations.

Equation 2 shows the calculation process for averages for each series of trials. The representation of the collective array of seconds is x, and n is the total number of trials, which in this case always amounted to 15. All cells located in the second column of both tables had this process applied, with their respective arrays of float seconds as parameters.

$$P_{\mathrm{n}} = 100 \cdot \left(\frac{A_n}{A_{n-1}} - 1\right) \qquad \text{where n} \geq 1 \text{ and } A_{n-1} > 0$$

Equation 3. Underflow/Overflow percentage equation.

Equation 3 shows the calculation process for the underflow or overflow of the percentage. 'A' denotes the array of averages, and n indicates the index of the current trial under calculation. This process ratios the current average to the previous one, then decreases this ratio by a whole, and finally scales it to a percentage. This equation functions only for subsequent trials, since calculating improvement or degradation requires at least one previous trial. Moreover, due to having a division operator, the previous trial's average cannot amount to 0, which would mean that the earlier trial has all xi equal to 0. This metric corresponds to the overall average improvement between series of trials. The goal for this metric involved achieving a negative value (underflow), which indicated a reduction in evacuation time.

5.2 Qualitative results

The qualitative results separate into two sub-sections: survey responses and observations recorded during trial simulations. While the survey responses are sufficient for answering the research questions, the recorded observations assist in identifying any unexpected weaknesses and threats.

Figure 9 displays a stacked bar chart for the survey question responses. The first step in creating this bar chart was exporting CSV files containing the responses in numerical form and fetching them from the Google Forms survey. Each response for each question had an integer ranging from one to five, with one signifying the "Strongly Disagree", two meaning "Disagree", three signifying "Neutral", four indicating "Agree" and five signifying "Strongly Agree". Inputting this numerical data into the 2D array in a Python script using Matplotlib enabled the creation of a bar chart.

5.2.1 Survey responses

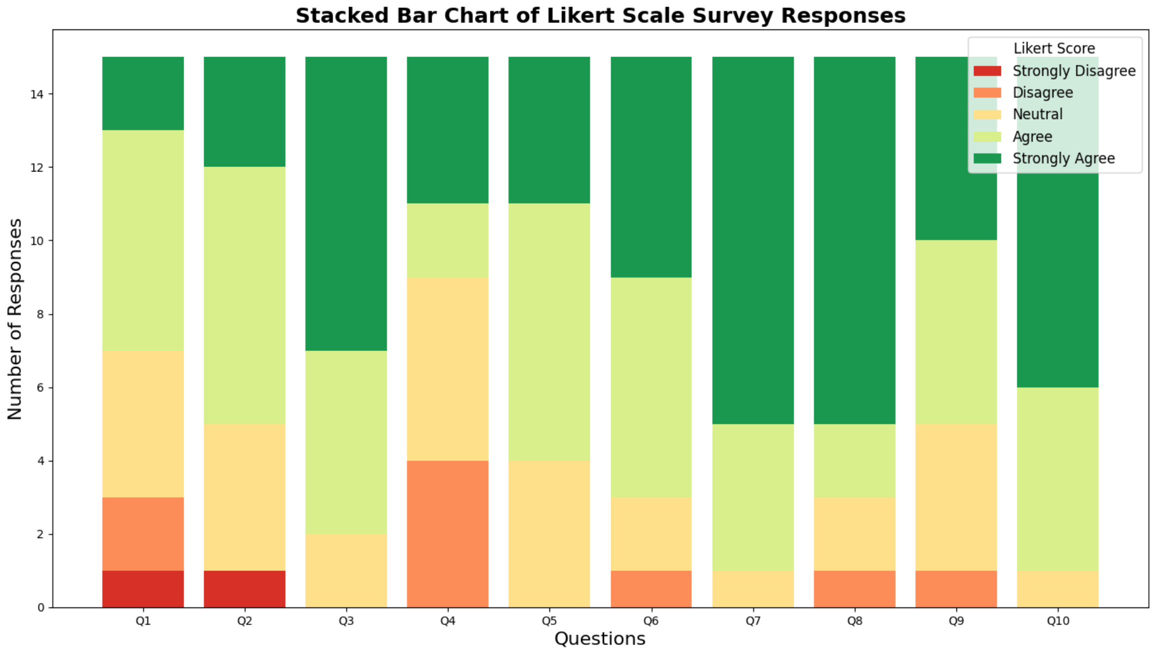


Figure 9. Stacked bar chart of Likert scale survey responses.

Figure 9: Bar chart, y-axis represents the number of responses, which always amounts to 15, meaning one response per participant. The x-axis represents the 10 questions asked in the survey (Full questions are in Appendix 2). The third component in the bar chart is the stacked bars, where each segment colour represents a response choice. The response choices with the collating colour codes are Strongly Disagree is red, Disagree is orange, Neutral is yellow, Agree is light green, and Strongly Agree is dark green. This bar chart displays all the data meaningfully, outlining the main weaknesses and strengths of the simulations from a user perspective.

Question Number	Average Rating
1	3.4
2	3.73
3	4.4
4	3.4
5	4
6	4.13
7	4.6
8	4.4
9	3.93
10	4.53

Table 4. Survey questions rating means.

Utilising Equation 2 but substituting x for the array for the numeric question responses mentioned above resulted in the average rating metric. All 10 questions with numerical data went through this process. Table 4 displays the individual ratings in its second column and their respective question numbers in the first column. This calculated metric provides a collective insight into what the average participant observed from the simulations.

5.2.2 Observations and comments

While the survey provided sufficient information regarding the simulations' strengths and weaknesses, the observations and participant comments provided insight into future opportunities for improvements and unexpected threats of the prototype. The notes for observations and recurring comments used a condensed version of the participant's alias but still maintained their uniqueness. Table 5displays the mentioned comments and observations.

Participant Alias	Observation / Comment
Doze	Break taken due to nausea from motion sickness. Participant explained that it may be due to the running quicker during second trail.
Hump	Participant suggested creating NPCs that evacuate with a future user to further enhance situational realism.
Fade	Participant felt slight nausea, particularly when traversing the fire exit stairs section.
Steep	Participant mentioned an issue when turning as the USB wire connecting the VR headset to the PC would restrict any excessive turning, more specifically when physically turning more than 180 degrees.
Proof	Participant explained how a different route to another fire exit would be faster, but it would not be the safest one as it conflicts with other evacuation routes. Addition of warnings would solve this issue.
Blot	Likewise, to previous comments, the mentioning of nausea induced from stairs section.
Ship	Like the previous set of trials nausea was induced from fire exit stairs section. Discussed and agreed participant that was due to the spiralling structure and small area to traverse.
Mocha	Participants got confused about where to evacuate on the first trail. The path line clipped in a floor model and was not fully visible, confusing the participant. This issue did not repeat itself in the subsequent trials.

Table 5. Observations and Participant Comments from trials.

Four out of the fifteen participants (Doze, Fade, Blot, Ship) reported experiencing nausea, particularly when navigating the fire exit stairs, with two attributing this discomfort to the quick pacing or the spiral structure of the staircase. One participant (Steep) noted a technical limitation caused by the VR headset's USB cable, which restricted full physical turning and affected immersion. Additionally, one participant (mocha) described initial confusion due to a visual path line clipping through the floor, impacting their navigation ability. Other feedback included suggestions for improvement, such as adding non-player characters (hump) to increase realism and implementing warnings for unsafe routes (proof). Table 5 highlights usability challenges and constructive suggestions, providing valuable insights for refining the VR simulation and enhancing user experience.

5.3 Discussion

Evaluation of Table 4 resulted in the conclusion that the system is effective in reducing evacuation time through subsequent trials. This reduction means that participants managed to gain knowledge of the evacuation route. Furthermore, although the third trial was unassisted, the overall evacuation time decreased by 3%, further demonstrating participants' retention of the evacuation route. Figure 8, provided the means for comparison of the evacuation time histograms. The first and third series of trials showed that, in the third series, no participants took more than 1.5 minutes to evacuate the virtual Xamk campus in Kotka. Moreover, the first evacuation series has a left-skewed distribution. This pattern means that it was a common occurrence during the first series that most participants would take longer to evacuate compared to the minority. The subsequent evacuation trial revealed a shift to a right-skewed distribution pattern. This shift further shows the participants' knowledge retention during simulation evacuations. The next step was comparing the histogram of the assisted second series of trials to the unassisted third series of trials. The data showed improvement, with the proportion of participants taking 30–45 seconds for evacuation increasing from 27% (0d.p.) to 40% (0d.p.). The information in Table 2 showed that evacuation time improvement decreased over subsequent trials, as participants approached their optimal evacuation time. Therefore, further trials might assist participants who took longer to evacuate, but improvement may be negligible in participants who are already effective in evacuating.

Table 3 underflow percentages evaluation concluded that there was the same pattern of improvement for the response times as seen in the evacuation times. This pattern demonstrated knowledge retention and boosts confidence when participants made decisions during emergencies. Table 4, which presents the survey rating for question three, further supports this statement. This question aimed to give insight into the decision-making process and their confidence in their decisions. Figure 9 shows that 53% of participants felt noticeably confident in their choices, while 13% reported feeling neutral about their decisions. The last 34% of participants, while not as satisfied with their decision as the majority, still felt confident in their choices. This distribution demonstrated VR fire evacuation simulations' ability to impact users and train them to make quicker and more confident decisions. It is essential to remember that if the game environment does not accurately represent the real-world counterpart in scale, positioning, rotation, and structure, it could endanger users in a real-world emergency.

While both sets of results ranged from neutral to positive, the survey also collected disagreements. Although the simulation closely mimicked real-life situations, it could never fully replicate the stress and urgency of an actual evacuation. Moreover, examination of question four further supported the idea that immersion does not necessarily mean replicating a real situation. Implementing new elements such as Nav-mesh NPCs and dynamic fire systems could further enhance these results. However, this fell outside the study’s scope and would have also reduced the prototype’s performance.

Questions five and ten of the survey assessed the overall experience and opinions of the participants. Question five provided positive results, although 47% (0d.p.) only agreed and did not strongly agree. This response percentage indicated that the majority may have seen the value of VR evacuation simulations in providing evacuation training but may have not wanted to rely on the simulations completely. The qualitative results established a new concept that combines traditional fire drills with VR simulations to enhance users’ confidence in evacuating premises during emergencies. Question ten further supported this concept, as 47% of participants would have suggested the training prototype to other inexperienced users. The high positive ratings of these questions displayed the competency of such a system. Further examination suggests the usage of the system as an enhancement for fire evacuation training, rather than as a replacement.

Questions six and nine considered the participants' interaction with the prototype during runtime. While both received positive response ratings, the results again indicated that improving user interaction and the GUI could have created a more informative simulation experience. Participant Proof mentioned that adding real-time warnings and signs of danger could have boosted the user experience. This enhancement would have increased the knowledge gained from trials and helped ensure no misunderstandings regarding the evacuation route, since the shortest path is not always the safest. Adding NPCs to display the number of evacuees using each emergency exit can help mitigate such misjudgements.

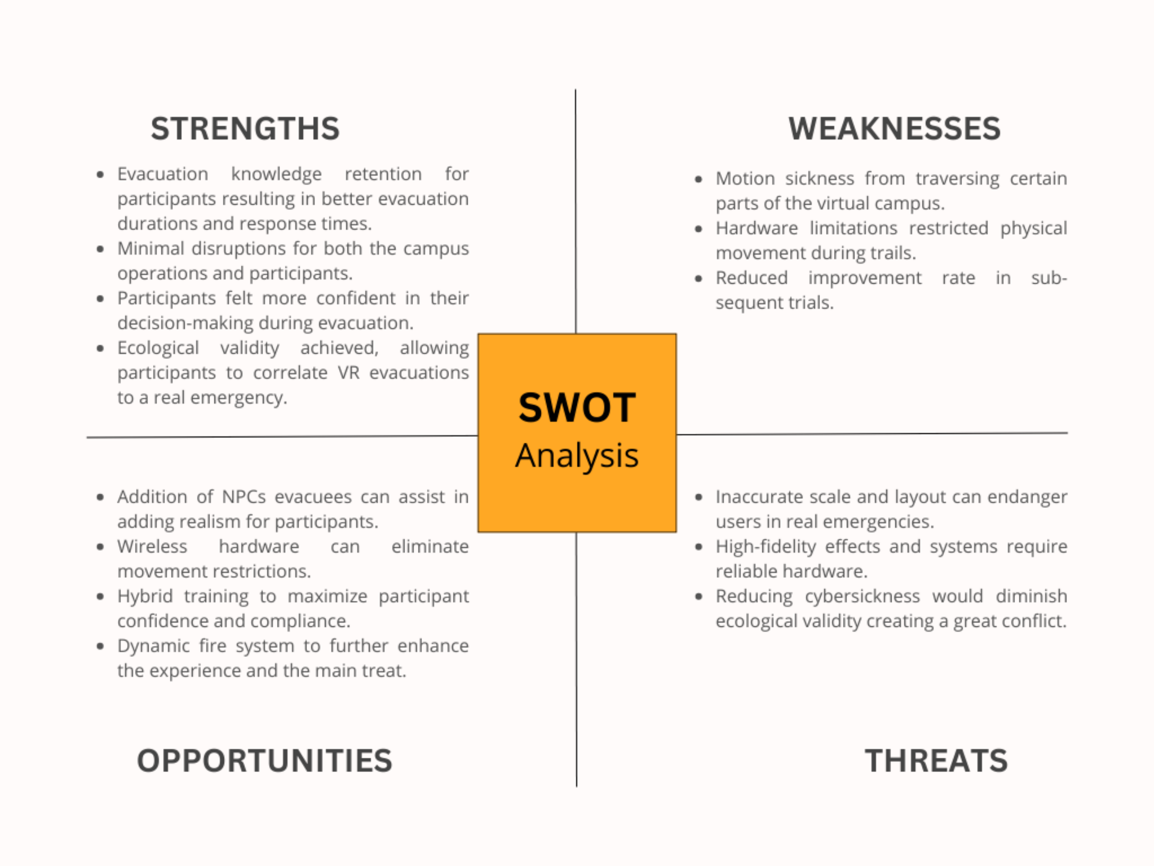
Question seven focused on the disruption in the participants' schedules. This question identified one of the greatest strengths that VR fire evacuation simulations yield, as 67% of participants found the VR simulations were less disruptive than conventional fire drills. This reduced disruption is the main selling point of VR evacuation simulations, as, due to fewer disruptions, participants were more compliant and less avoidant of evacuation training.

Question eight directly provided insight regarding ecological validity from a user's perspective. This question had 67% of participants strongly agree with the Likert statement, which displayed the effectiveness of using LiDAR FBX models to replicate real-world structures. Although a small number of participants felt neutral or disagreed with the statements, this could have been a result of reduced graphics and texture due to limited hardware, lack of rendering knowledge, and time restrictions.

This study identified hardware limitations, such as how a tethered headset restricted physical movement and broke immersion. As noted from participant comments, standalone devices could have mitigated this issue at the cost of performance and a lack of hardware variety for the system. These aligned with prior work emphasising the trade-offs between ecological validity and performance optimisation in VR. Although the team initially planned to produce a standalone prototype, they opted for a more suitable version that accommodated time restrictions and limited hardware.

The observations identified that a minimum of 14% of participants experienced motion sickness and nausea, which participants attributed to environmental configuration issues. Teixeira, J., & Palmisano, S. (2021) explain how a limited FOV induced cybersickness for VR participants. Furthermore, cybersickness intensified through sequential trials. This threat pointed out a new factor: the conflict between ecological validity and VR game design.

With consideration to the above, this thesis presents a SWOT analysis outlining the strengths, weaknesses, opportunities, and threats of VR fire evacuation simulations. Figure 10 displays the mentioned analysis.



STRENGTHS

- Evacuation knowledge retention for participants resulting in better evacuation durations and response times.
- Minimal disruptions for both the campus operations and participants.
- Participants felt more confident in their decision-making during evacuation.
- Ecological validity achieved, allowing participants to correlate VR evacuations to a real emergency.

WEAKNESSES

- Motion sickness from traversing certain parts of the virtual campus.
- Hardware limitations restricted physical movement during trails.
- Reduced improvement rate in sub-sequent trials.

OPPORTUNITIES

- Addition of NPCs evacuees can assist in adding realism for participants.
- Wireless hardware can eliminate movement restrictions.
- Hybrid training to maximize participant confidence and compliance.
- Dynamic fire system to further enhance the experience and the main treat.

THREATS

- Inaccurate scale and layout can endanger users in real emergencies.
- High-fidelity effects and systems require reliable hardware.
- Reducing cybersickness would diminish ecological validity creating a great conflict.

Figure 10. VR fire evacuation simulation SWOT analysis

In summary, the VR fire evacuation simulation demonstrated clear strengths in improving participant preparedness, confidence, and training efficiency, while offering a high degree of ecological validity and minimal disruption compared to traditional drills. However, challenges such as motion sickness, hardware limitations, and specific design flaws highlight areas for further refinements. Opportunities exist to enhance realism and accessibility through integrating NPCs, wireless hardware, and hybrid training approaches. However, one must balance these against threats related to ecological validity, system performance, and participant variability.

6 CONCLUSION AND RECOMMENDATIONS

6.1 Conclusive summary

This study aimed to develop and evaluate a VR fire evacuation simulation tailored for the XAMK Kotka campus. The primary objective was to enhance emergency preparedness and minimise operational disruptions. By leveraging immersive technology, the investigation sought to address gaps in traditional training methods, particularly the infrequency of drills and their inability to replicate high-stress scenarios.

The theoretical framework reviewed the literature and work already done on this topic. The review introduces new subtopics such as environmental accuracy, immersive learning, and the viability of simulation-based training. Papers have analysed these topics and identified factors such as ecological validity and VR immersion, which informed the creation of the final prototype Having anchored the prototype's design in validated theories, the study ensures that its findings contribute to XAMK's safety protocols and the broader discourse on VR's role in emergency preparedness.

The methodology employed a mixed-methods approach, combining quantitative data collection through the Unity Experiment Framework (UXF) with qualitative feedback from participant surveys and observational notes. A LiDAR-scanned 3D model of the campus ensured ecological validity, while Unity's VFX Graph and OpenXR toolkit facilitated the creation of dynamic fire effects and VR interactions.

Participants completed three sequential trials, two assisted and one unassisted, with metrics such as evacuation durations and response times logged and saved automatically. Post-simulation surveys using a 5-point Likert scale provided insights into user perceptions and outlined issues with the prototype. The mixed approach enabled evaluation of the system through practical measurements and subjective assessments, providing a comprehensive understanding of its strengths and limitations.

Key findings revealed significant improvements in participant performance across sequential trials. Evacuation durations decreased by 24%, while response times improved by 40%, demonstrating VR fire evacuation simulations' ability to enhance evacuation skills and decision making. The third trial, conducted without guidance, still produced a 3% reduction in evacuation times, highlighting knowledge retention. However, the results also showed that reductions in evacuation and response times decrease significantly over multiple trials. Survey results reinforced these outcomes, where 67% of participants strongly agreed that the simulation had replicated real-world conditions (Q8) and 53% expressed high confidence in their decision-making (Q3). However, motion sickness affected 14% of users, particularly during stair navigation, and hardware limitations (e.g., tethered headsets) occasionally disrupted immersion. These findings collectively validated VR's potential while emphasising the need for iterative refinements. Strengths included high ecological validity, demonstrated by the LiDAR-accurate environment, and the prototype's scalability through modular Unity prefabs. Weaknesses centred on technical trade-offs, such as simplified fire dynamics to maintain performance on accessible hardware, and recurring issues like path-line clipping. Opportunities emerged for crowd simulation via non-player characters (NPCs) and wireless headset integration to eliminate cable restrictions. Conversely, threats included the risk of over-reliance on VR-specific tools, which could hinder future engine migrations.

6.2 Future works

The VR-based fire evacuation simulation developed for XAMK's Kotka campus has demonstrated promising results in improving evacuation preparedness, yet several limitations have presented opportunities for future research. An area for investigation is the integration of computational fluid dynamics to simulate heat-driven smoke, addressing the current restriction on simplified particle systems and enhancing realism. Expanding the simulation to include dynamic crowd behaviour via non-player characters (NPCs) would address the lack of realistic interactions during evacuations. Further studies should also prioritise hardware accessibility by adopting wireless VR headsets to eliminate cable restrictions and testing standalone devices to ensure scalability across institutions with limited resources. Addressing cybersickness, reported by 14% of participants, could involve implementing adaptive locomotion methods (e.g., teleportation), optimising field-of-view settings, and narrow and non-linear area navigation. Moreover, studies assessing skill retention over months or years could validate the long-term efficacy of VR training compared to traditional drills. These advancements could transform VR into a cornerstone of holistic safety protocols, bridging the gap between controlled simulations and real-world emergency responsiveness. These advancements could transform VR into a cornerstone of safety protocol training, bridging the gap between controlled simulations and real-world emergency responsiveness. Future work should aim to balance technical innovation with user-centric design, ensuring VR's role as both a training tool and a procedural improvement in fire safety.

REFERENCES

Bilotta, F.F., Werner, S.M., Bergese, S.D. & Rosa, G. 2013. Impact and implementation of simulation-based training for safety. *The Scientific World Journal*, 2013(1), 652956. DOI:10.1155/2013/652956. [Accessed 25 April 2025].

Birckhead, B., Khalil, C., Liu, X., Conovitz, S., Rizzo, A., Danovitch, I., Bullock, K. & Spiegel, B. 2019. Recommendations for methodology of VR clinical trials in health care by an international working group: Iterative study. *JMIR Mental Health*, 6(1), e11973. DOI:10.2196/11973. [Accessed 25 April 2025].

Bourhim, E.M. & Cherkaoui, A. 2020. Efficacy of VR for studying people's pre-evacuation behavior under fire. *International Journal of Human-Computer Studies*, 142, 102484. DOI: 10.1016/j.ijhcs.2020.102484. [Accessed 25 April 2025].

Brookes, J., Warburton, M., Alghadier, M., Mon-Williams, M. & Mushtaq, F. 2020. Studying human behavior with VR: The Unity Experiment Framework. *Behavior Research Methods*, 52(2), 455–463. DOI:10.3758/s13428-019-01242-0. [Accessed 21 February 2025].

Bruppacher, H.R., Alam, S.K., LeBlanc, V.R., Latter, D., Naik, V.N., Savoldelli, G.L., Mazer, C.D., Kurrek, M.M. & Joo, H.S. 2010. Simulation-based training improves physicians' performance in patient care in high-stakes clinical setting of cardiac surgery. *Anesthesiology*, 112(4), 985–992. DOI:10.1097/ALN.0b013e3181d3e31c. [Accessed 23 April 2025].

Bugli, D., Dick, L., Wingate, K.C., Driscoll, S., Beck, D., Walsh, B. & Greiner, A.L. 2023. Training the public health emergency response workforce: A mixed-methods approach to evaluating the VR modality. *BMJ Open*, 13(5), e063527. DOI:10.1136/bmjopen-2022-063527. [Accessed 26 April 2025].

Chittaro, L. & Sioni, R. 2015. Serious games for emergency preparedness: Evaluation of an interactive vs. a non-interactive simulation of a terror attack. *Computers in Human Behavior*, 50, 508–519. DOI: 10.1016/j.chb.2015.04.033. [Accessed 25 April 2025].

Dakeev, U., Pecen, R. & Yildiz, F. 2020. Development of VR simulation for emergency evacuation in education. *In: 2020 IEEE Frontiers in Education Conference (FIE), 21–24 October 2020, Uppsala, Sweden. IEEE*, 1–5. DOI:10.1109/FIE44824.2020.9274110. [Accessed 25 April 2025].

Guo, Y., Li, S., Liu, H., Chen, R., Zhao, X. & Zhang, Q. 2020. Enhancing VR fire safety training through stress arousal and cognitive load modulation. *Fire Technology*, 56(5), 2163–2189. DOI:10.1007/s10694-020-00976-0. [Accessed 21 February 2025].

Guzmán, D.E., Rengifo, C.F., Guzmán, J.D. & Garcia Cena, C.E. 2024. VR games for cognitive rehabilitation of older adults: A review of adaptive games, domains and techniques. *VR*, 28(2), 92–117. DOI:10.1007/s10055-023-00800-8. [Accessed 25 April 2025].

Gwynne, S., Kuligowski, E., Boyce, K., Nilsson, D., Robbins, A., Lovreglio, R. & Thomas, J. 2020. Enhancing egress drills: Preparation and assessment. *Fire Safety Journal*, 120, 103041. DOI: 10.1016/j.firesaf.2020.103041. [Accessed 25 April 2025].

Gwynne, S., Galea, E.R., Owen, M., Lawrence, P.J. & Filippidis, L. 2016. A review of the methodologies used in evacuation modelling. *Fire and Materials*, 40(1), 113–137. DOI:10.1002/fam.2326. [Accessed 21 February 2025].

Kinateder, M., Ronchi, E., Nilsson, D., Kobes, M., Müller, M., Pauli, P. & Mühlberger, A. 2014. VR for fire evacuation research. *Annals of Computer Science and Information Systems*, 2(4), 313–321. DOI:10.15439/2014f94. [Accessed 23 April 2025].

Ruesseler, M., Weinlich, M., Müller, M.P., Byhahn, C., Marzi, I. & Walcher, F. 2012. Republished: Simulation training improves ability to manage medical emergencies. *Postgraduate Medical Journal*, 88(1040), 312–316. DOI:10.1136/postgradmedj-2011-130748rep. [Accessed 26 April 2025].

Saghafian, M., Laumann, K., Akhtar, R.S. & Skogstad, M.R. 2020. The evaluation of VR fire extinguisher training. *Frontiers in Psychology*, 11, 593466. DOI:10.3389/fpsyg.2020.593466. [Accessed 24 April 2025].

Salas, E., Wildman, J.L. & Piccolo, R.F. 2009. Using simulation-based training to enhance management education. *Academy of Management Learning & Education*, 8(4), 559–573. DOI:10.5465/amle.8.4. zqr559. [Accessed 26 April 2025].

Schmuckler, M. 2001. What is ecological validity? A dimensional analysis. Division of Life Sciences, *University of Toronto at Scarborough*, 2(4), 419–436. [Accessed 21 February 2025].

Slater, M., Spanlang, B. & Sanchez-Vives, M.V. 2009. Inducing virtual body ownership in an immersive virtual environment. *Presence: Teleoperators and Virtual Environments*, 18(4), 294–303. DOI:10.1162/pres.18.4.294. [Accessed 04 May 2025].

Slater, M., Rovira, A., Southern, R., Swapp, D., Zhang, J.J., Campbell, C. & Levine, M. 2013. Bystander responses to a violent incident in an immersive virtual environment. *PLoS One*, 8(1), e52766. DOI: 10.1371/journal.pone.0052766. [Accessed 24 April 2025].

Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. DOI:10.1016/0364-0213(88)90023-7. [Accessed 04 May 2025].

Teixeira, J. & Palmisano, S. 2021. Effects of dynamic field-of-view restriction on cybersickness and presence in HMD-based VR. *VR*, 25(2), 433–445. DOI:10.1007/s10055-020-00454-9. [Accessed 04 May 2025].

Yu, S. 2022. Application of VR intelligent technology in fire evacuation simulation of urban rail transit station. *Procedia Computer Science*, 208, 223–230. DOI: 10.1016/j.procs.2022.10.033. [Accessed 26 April 2025].

Zhu, X. & Li, Y. 2021. A critical review of virtual and augmented reality applications in emergency management for built environments. *Automation in Construction*, 132, 103956. DOI: 10.1016/j.autcon.2021.103956. [Accessed 04 May 2025].

A1. APPENDIX 1. PROTOTYPE LINK

GitHub link: https://github.com/ColtonSammut/Unity_Thesis_GCOSA001.git

A2. APPENDIX 2. LIKERT SCALE SURVEY QUESTIONS

1. The VR simulation made me feel as if I was experiencing a real fire emergency in the campus environment.
2. The simulation prompted me to behave as I would in a real fire emergency (e.g., urgency, decision-making).
3. I felt confident in my ability to evacuate the campus safely after completing the VR simulation.
4. I felt more aware of my surroundings (e.g., exits, obstacles) in the VR simulation than during conventional fire drills.
5. I believe VR-based training can be a suitable replacement or supplement for conventional fire drills.
6. I found it easy to navigate through the virtual campus environment using the provided controls.
7. The VR drill was less disruptive to my normal activities compared to traditional fire drills.
8. The VR environment realistically represented the layout and features of the actual campus.
9. The simulation provided clear instructions and feedback during the evacuation drill.
10. I would recommend VR fire evacuation training to other campus occupants.