Lost In Wilderness

Sincere Thanks to our Faculty: Dr.Sharad Sharma.

Johannas Katikala
The Anuradha and Vikas Sinha
Department Of Data Science
University Of North Texas
Johannasmeelakatikala@my.unt.edu

Sahana Loka
Department Of Information Science
University Of North Texas
Sahanaloka@my.unt.edu

Abhaya Ladalla
Department Of Information Science
University Of North Texas
Abhayaladalla@my.unt.edu

***Abstract*—In real-life wilderness survival experiences, quick decision-making, resource management, and adaptability are important, while traditional survival training is dangerous and not for the masses. Survival in the Wilderness, this project suggests, is an experiential Virtual Reality (VR) simulation created using Unity with a focus on providing a safe, interactive, and immersive environment in which users learn and practice wilderness survival techniques. Set in a realistic forest environment, the game involves resource gathering, hunger and thirst management, fire creation, cooking, and defending against AI-powered wild animals. The simulation focuses on the necessity of environmental awareness and decision-making under stress. The project beneficiaries include teachers teaching survival skills, emergency readiness groups training members virtually, and game players interested in interactive and educational games. By combining gamified interaction and real-world learning, this VR environment offers a new approach to survival strategy learning.**

Keywords—Virtual Reality, Wilderness Survival, Unity Game Development, Resource Management, Immersive Learning, Gamified Training, Environmental Awareness, AI Interaction, Emergency Preparedness, Experiential Simulation)

I. Introduction

Virtual Reality (VR) is an efficient tool for simulation learning and experiential education over the past few years. Survival in the Wilderness project has envisioned how to utilize it by creating a VR software that mimics real survival situations in an actual forest setting. This game, built in Unity, provides the learner with a tough but safe setting in which the student is able to carry out survival tasks such as gathering resources, hunger and thirst levels, campfires, cooking, and keeping aggressive wildlife at bay. Its purpose is to teach basic thinking, conservation, and tactical thinking through being as realistic as possible to actual environments.

A. Goals and Objectives

The major objectives of the project are:

- To model wilderness survival situations within a VR setting.
- To design an experiential learning device that uses game play and instructional content.
- To encourage resource allocation and strategic decision-making under stressful conditions.
- To deliver a safe and reproducible training system for readiness teams and instructors.
- To enable greater user immersion with AI-based threats and real-time feedback systems.

The final objective is to identify how gamified virtual environments can increase user engagement and retention of survival training compared to traditional training. Unlike traditional training, this simulation is repeated several times over, can be adjusted to fit any level of expertise, and utilized without risk of outdoor survival training exercises.

B. Modeling the VR Environment

The virtual environment was modeled after a real-world wilderness scenario. The forest has different features such as dense underbrush, rocks, running lakes, terrain elevation, animal trails, and survival equipment. These were modeled using Google SketchUp and Blender and exported to Unity for use.

The design elements are:

- Natural Landscapes: Rocks, trees, grassland, and lakes create a real outdoor environment.
- Survival Inventory: Interactive items such as axes, water bottles, meat, firepits, and lighters are included.
- Wildlife and AI Enemies: Realistic wild animals modeled and animated use Unity's NavMesh system to generate smart movement and attacking behavior.

Textures and shaders are used to produce realism—wood grain on trees, reflective water surfaces, fire effects, and ambient sound effects such as birds and wind create immersion. Animations like tree swaying, crackling fire, animal movement, and player gesture upon interaction are provided.

C. Application Usage and Interactions

The Wilderness Survival app presents a natural, first-person experience of a highly interactive virtual forest environment. Players undergo a set of survival tasks that mimic real tasks in the wilderness. In simulation, they are able to chop down trees for wood by using a virtual axe, break rocks to collect stones, and engage with water bodies to drink directly or fill water bottles. Fire-crafting is a key survival element, in which players have to use a lighter to activate a firepit, which the player then uses to cook raw meat to restore health. Combat elements are implemented in the game through AI-powered wild animals that respond to player proximity, for which players will need to use bows and arrows to fend them off. Heads-up display (HUD) enables players to monitor essential survival statistics such as health, hunger, and thirst, encouraging careful resource management.

The software is targeted to a broad range of users, from students and teachers watching survival techniques to emergency readiness groups seeking simulation-based training software to game players seeking entertaining, narrative-based survival games. Blending education and entertainment, Survival in the Wilderness demonstrates that computer simulations hold the promise of providing valuable learning without risk of exposure in the wilderness. The environment promotes flexibility and decision-making, yet as an experiential training and skill-learning platform.

II. Related Works

Virtual Reality (VR) environments have been increasingly applied to immersive simulations, learning, training, and game play. We drew upon prior scientific research in VR horror, survival game mechanics, adaptive gameplay, gamified training, and intelligent NPC behavior in developing our VR survival game. Below, we outline some of the pertinent scientific research that guided our project.

Lin (2020) in "Fear in Virtual Reality (VR): Fear Elements, Coping Reactions, Immediate and Next-day Fright Responses Toward a Survival Horror Zombie Virtual Reality Game" examined participants' psychological reactions while playing interactive VR horror games. Distinguishing between place illusion (PI) and plausibility illusion (PSI), the study showed that PSI elements—those concerning the situation's plausibility—elicited more fear and coping reactions. Taking cue from this, our VR survival game is based on creating high PSI environments to instill emotional engagement and strategic action among players. Ulmer et al. (2022) in "Gamification of Virtual Reality Assembly Training: Effects of a Combined Point and Level System on Motivation and Training Results" investigated gamification elements, i.e., points and levels, in VR assembly training.

Their findings indicated that gamified VR training not only increased motivation but also led to enhanced real-world performance results. Consistent with this, our VR survival game has progression systems and feedback to keep players motivated and learning through challenge. De Lima, Silva, and Galam (2023) proposed a machine learning-based approach in "Adaptive Virtual Reality Horror Games Based on Machine Learning and Player Modeling" to personalize VR horror experiences. Their approach personalized game content based on personal fears of players and thus increased immersion and emotional impact. This reminds us to include adaptive capabilities in subsequent versions of our survival game to better match each player's emotional responses.

Scorgie et al. (2022) in their meta-review "Virtual Reality for Safety Training: A Systematic Literature Review and Meta-Analysis" showed that safety training via VR is much more effective than traditional means in retention of knowledge and interaction. This proves the deployment of VR survival games such as ours as not only a source of entertainment but also training of cognitive and strategic skills such as quick decision-making, risk assessment, and allocation of resources.

Strannegård et al. (2024) in "Survival Games for Humans and Machines" compared human and artificial intelligence agent survival performance on unpolluted survival tasks. It highlighted how and why such a feature as perception, memory, and reward configuration impacts survival behaviors. Referencing the same, our VR game positions players into survival strategic decision-making to move about in an unusual and changing new world.

Herumurti et al. (2020) designed a VR-based FPS survival maze in "Development of First-Person Shooter Game with Survival Maze Based on Virtual Reality". The research showed that combining 3D NPCs, VR immersion, and survival features was effective for creating quality gameplay. Our VR survival game takes this further by developing a first-person game with interactive survival capabilities and intelligent NPC behavior.

Singh and Kaur (2022) in "Game Development Using Unity Game Engine" presented an overview of the capability of Unity to design interactive worlds. Unity's physics engine, rigid body physics, and massive base of developers are some of the aspects that render Unity an ideal platform to develop on for VR. All these aspects were necessary in developing our VR survival game with realistic interactions and efficient development procedures.

Finally, Meliones and Plas (2017) in "Developing Video Games with Elementary Adaptive Artificial Intelligence in Unity: An Intelligent Systems Approach" emphasized the significance of adaptive AI as far as increasing the responsiveness of video games to player actions is concerned. Their development of elementary adaptive AI systems made our plan to create NPCs that respond dynamically to player action in future updates the ultimate objective, bringing realism and challenge to the survival world.

In summary, the reviewed literature establishes the significance of immersion, flexibility, motivation, and realism in VR. Our VR survival game project integrates these research findings to offer a considerable and interactive experience, which is the basis for future enhancement through player modeling and adaptive AI.

III. Implementation

System Implementation Flow and Structure:

The below diagram illustrates the step-by-step sequential process undertaken in the development of the VR-based survival game. It begins with 3D modeling and exportation and importation of assets to the Unity engine. It continues to scripting, addition of behavior, and addition of essential elements of gameplay. It continues to show the interaction between the user and the VR client and ends at the testing, debugging, and optimization stage. The step-by-step sequential process enables a sequential and systematic process of development.

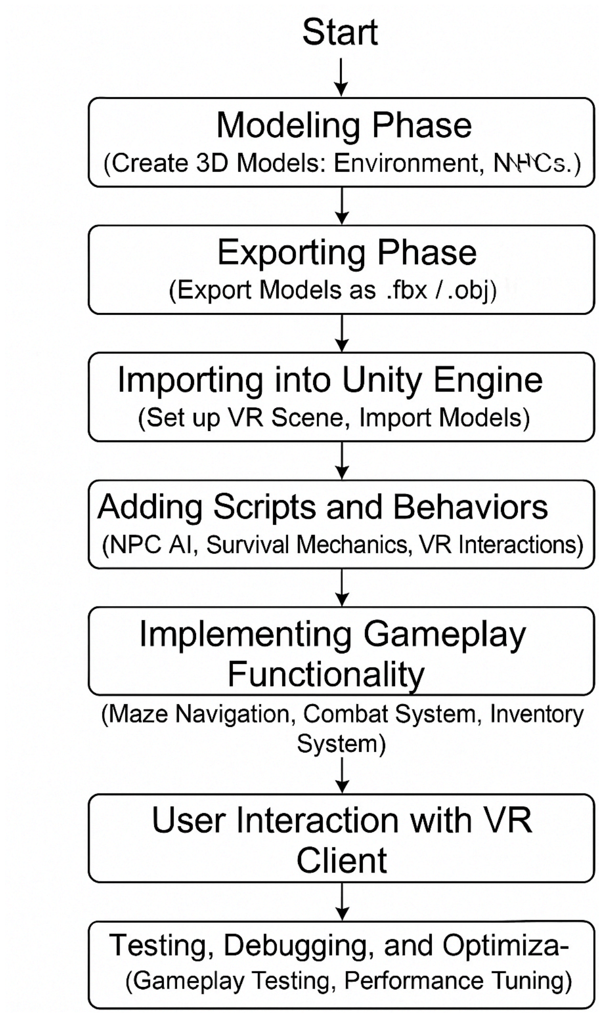

Figure 3.1: System Architecture

A. Modelling

The initial stage of implementing the game involves developing the visual environment that includes forests, lakes, rocks, trees and firepits, and other assets that populate the game world. Creating high-quality visual models through 3D modelling software requires Blender or SketchUp during this phase.

a) *Environment Assets:* The main environment elements, including trees and rocks, and lakes, were modeled according to correct sizes, and their surfaces received appropriate levels of detail. The trees display authentic characteristics because of their natural bark texture. The stone surfaces display rough textures, whereas lake waters show full transparency, and firepits present arranged stones in a circular pattern, which serves as space for fire visual effects.

b) *Low vs High Poly Models:* The rocks, trees, and other background elements are modelled in low-poly for efficiency.

c) *Exporting and Importing to Unity:* After completing the models, they are exported into standard formats like .fbx or .obj and imported to Unity. The textures and prefabs are created in Unity to fit into the game's overall visuals.

d) *Terrain Building:* The Unity terrain tool is used to create a terrain landscape.

B. Game Mechanics

The Game Mechanics constitute gameplay elements because they enable resource gathering together with survival management, and player actions with C# as a programming language. Implementation of tree cutting and rock smashing occurs through setting numeric values to trees and rocks. The game allows players to gather resources by using an interaction button as they approach collectable items and store them in their inventory.

A feature of Firestarter mechanics enables players to ignite wood materials they have gathered during exploration. The game verifies player possession of needed materials, and after verification, it activates fire animations through particle systems.

The Survival systems evaluate the health condition of the player while also examining hunger levels and hydration. Health levels decrease when players encounter environmental dangers. Survival needs appear through sliders and timers on the user interface, while penalties like blurred vision or slowed movement happen when survival needs go unattended.

C. AI Behavior

A good survival game requires enemies and threats, which can be achieved through Unity's NavMesh system, enabling AI Navigation functionality. With NavMesh, the Unity system enables AI characters to avoid barriers when creating their movement pathways. Unity uses the navigation mesh concept to divide the terrain areas for AI characters into walkable and non-walkable sections.

The enemies take the form of both hostile human enemies and animal predators in multiple locations. The entities roam the area at random or rest until something prompts them to move. The detection system determines threats by monitoring spaces within a predetermined distance radius.

AI transforms into chasing mode from its idle and patrolling states. At short distances, enemies will strike the player, thus reducing their health bar levels. The enemy stops its patrolling when a player moves a sufficient distance away, which makes the enemy return to its previous behavior. Each AI character uses predefined movements that match their operational state.

D. Inventory Systems

The survival experience extends beyond acquiring items because it requires effective management skills. Through the inventory system, players can both obtain and safely keep their resources as well as utilize them. The interface uses basic grid elements to show item types via slots. Users can see item icons on the screen, and these icons present tooltips with quantity details along with usage instructions.

A single item acquisition in the world automatically removes it from its current location and places it into the proper inventory space. A single stackable item slot exists that enhances quantity. All distinct items receive their dedicated slot position. A button click operation activates items while script modules manage these items effectively. Tool and shelter creation through combining resources should be added at this time.

E. Interactions

Survival becomes more immersive because of its detailed interaction system. A player can seek a lake to fill an empty bottle through the quip feature. The system changes the bottle status from "Empty bottle" to "Full Bottle". Players can wear equipment that consists of a lighter tool for such activities. Clicking the "Use" button near firepits or dry bushes starts the fire triggered by this action. The system activates animations for lighter tools while utilizing triggering particle effects, which create flames.

The player can activate the bow equipped after completing the bow and arrow creation. The drawing action begins when the player holds the mouse buttons before the arrow appears, and the simulated, realistic trajectories start when the buttons are released through Unity's rigid body. Hitting enemies produces damage, but the player can retrieve any missed arrows that drop close to their position.

IV. FUNCTIONALITY

A. Vision

High -resolution textures and 3D models were integrated into the virtual forest environment. Rock, tree, lake, and survival equipment were modeled in SketchUp and Blender and imported into Unity. Bark textures, water bodies, and ground textures were optimized in Substance Painter to ensure that they are as realistic as possible. The graphical design was so immersive, and it offered players a realistic wilderness .

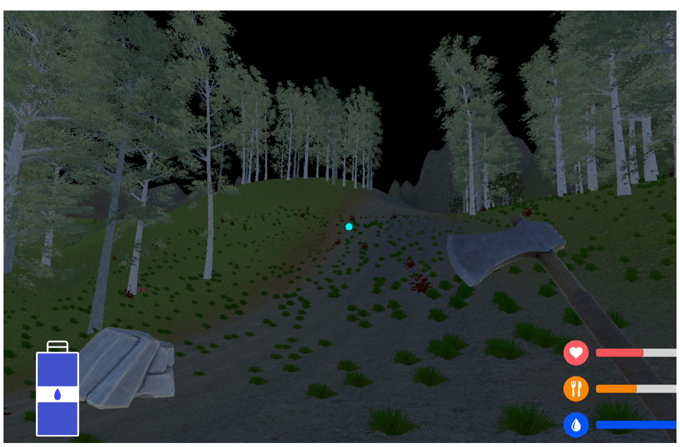

Fig.4 1. Forest Environment with Realistic Textures and 3D Models

B. Sound

Natural surroundings were simulated using ambient soundscapes like bird chirps, flowing rivers, the wind blowing through leaves, and distant growls of animals. Environmental sounds switched dynamically from day to night cycles. Sound effects were also added for player activities like cutting trees, lighter usage, or refilling water bottles.

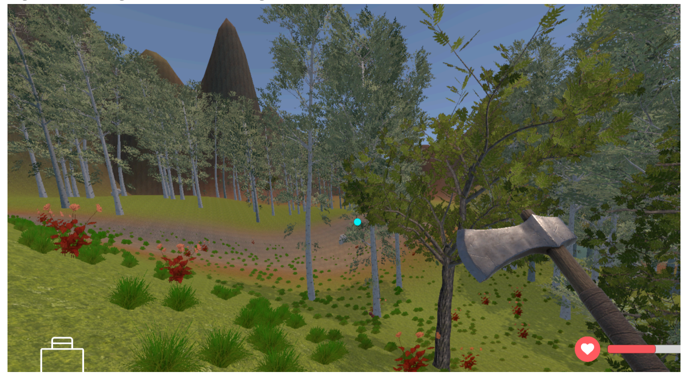

Fig.4. 2. Sound Mapping in Unity Environment

C. Animation

Multiple animated objects provided a sense of realism:

1. Fire Animation: Natural firepit flames and smoke particles that are sensitive to weather.
2. Water Animation: Ripples and reflections in lakes and rivers.
3. Animal Animation: Birds in flight, deer walking, and squirrels running realistically.

All the animations were controlled using Unity's Animator and Particle System assets

Fig.4. 3. Firepit Animation Showing Sparks and Smoke

D. Interactivity

The simulation has a minimum of five significant user-triggered events:

1. Tree Chopping: Player cuts wood with an axe.
2. Rock Breaking: Collection of stones by breaking rocks.
3. Water Collection: Drinking directly or refilling water bottles at lakes.

4. Fire Creation: Creating fires using gathered resources and a lighter.
5. Enemy Combat: Engaging with wild animals using arrows.

Fig. 4.4 Water Collection Mechanism with Interaction Prompt

E. Characters/Avatars

Non-Playable Characters (NPCs) like wolves and deer were used with path-following AI behavior using Unity NavMesh. These animated agents roam the globe and react to player proximity. Wolves attack and pursue players upon hunger level activation, creating challenge scenarios.

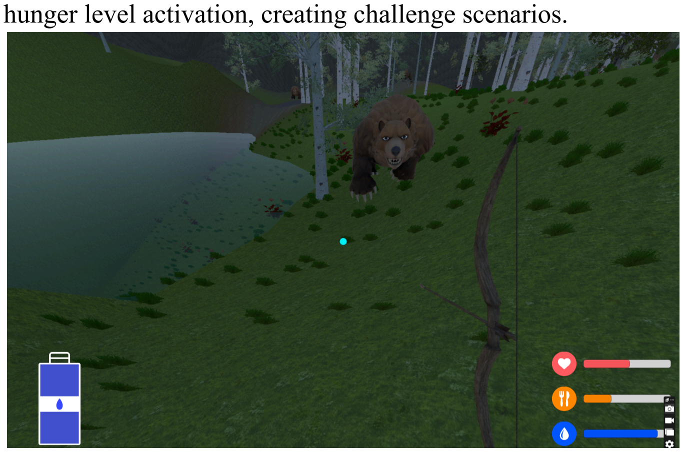

Fig.4. 5. Bear NPC Moving Along Path with NavMesh Agent

F. Sensors

Three different types of sensors were employed:

1. Proximity Sensor: Fires tooltips when the player approaches survival items.
2. Touch Sensor: Triggers item pickup when the player picks up objects.
3. Visibility Sensor: Some hidden objects (like a compass) are only revealed upon completion of some activities.

Fig.4. 6. Proximity Sensor Activation Around Survival Item(Lighter)

G. Player Controller

A First-Person Controller (FPC) was implemented to facilitate smooth navigation of the wilderness. Players can walk, run, crouch, and jump. Sprinting and stamina drain make survival more realistic.

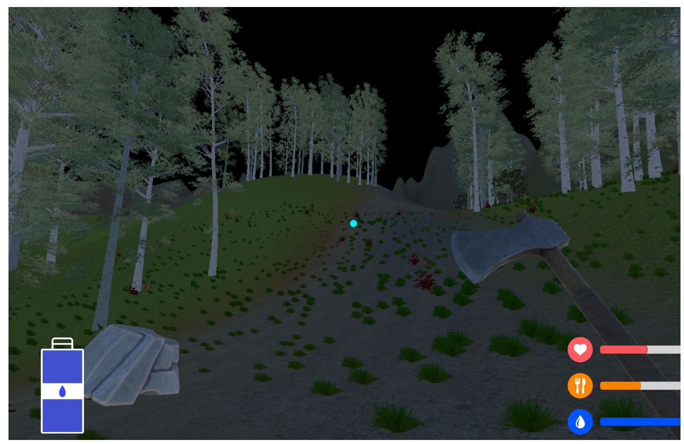

Fig. 4.7. First-Person Controller Setup in Unity

H. AI Implementation

AI-driven animals were used with:

1. Pathfinding Algorithms (Unity NavMesh)
2. Adaptive Behaviors: Wolves attack at night; deer flee from players.
3. Learning AI (Optional): NPC behavior can be changed based on past player interactions.

AI behaviors make the world seem dynamic and responsive to player action.


Fig. 4.8. AI Behavior Tree for Animal Response Logic

I. Interface Elements

1. Main Menu: Start, Settings, Exit.
2. Inventory UI: Shows collected wood, water, meat, etc.
3. Health & Status Bars: Track hunger, thirst, and level of health.
4. Feedback Panel: Display survival tips and warnings in a dynamic fashion.

Fig.4. 9. UI Elements

J.Mobile version

In addition to the VR version, there was also a mobile version of the game. It features:

- Lightweight optimized 3D models to ensure good performance on mobile phones.
- Touch controls used to handle movement, interaction, and combat.
- Reduced latency and easier navigation trimmed for smaller screen size.
- Survival mechanics are still present (hunger, thirst, health system).

This allows the users to enjoy the survival maze world without needing to use any VR hardware, hence making it accessible

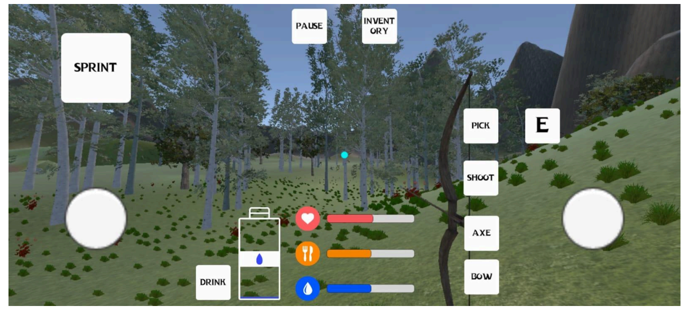

Fig. 4.10.Mobile Version

V. Conclusion

The workflow involved in making the VR survival game involved some very important steps like 3D modeling of environments, NPCs, and key objects; application of appropriate textures; rigging and character animation; and implementing interactive functionality by means of scripting and AI behavior code. Geometric accuracy was managed to keep performance low but yield realistic appearances. Textures were employed sparingly to give a realistic feel without putting a strain on system resources, and animations were introduced to give natural NPC movement and realistic user input.

The game can be played with a VR headset and controllers, where players navigate a survival maze, interact with the world, battle against hostile NPCs, and organize their inventory. A standard user manual will have commands like movement commands, combat tutorials, survival tips, and inventory commands to make sure that players are able to get the best out of the gameplay.

It is also an asset app as it not only provides recreation but also enhances logical thinking, decision-making, and stress management capabilities. Survival features mimic real-world real-life situation-solving processes to make the game helpful for users with learning and entertainment goals in mind.

Virtual reality is the most suitable technology to use in this context because it provides full immersion, which makes users believe that they are indeed inside the survival environment. VR has a greater emotional investment factor, increasing the appeal and simulation of survival missions compared to conventional flat-screen gaming.

Various issues were during the process, including performance optimization for VR (balance between high-fidelity models and framerate), NPC AI calibration to move naturally, and user comfort control to reduce VR sickness. There remain deficiencies in areas of AI behavior, richness being slightly restricted, certain nav glitching within mazes occasionally, and hardware dependency (requirement for a VR-enabled machine and headgear).

For future development, concepts are to include more sophisticated adaptive AI, adding multiplayer functionality, increasing the game world with procedural generation for greater replayability, and further refining for mobile VR platforms to make it more feasible.

Acknowledgment

We especially wish to thank Teaching Assistant Keerthana Srinivasan and Professor Dr. Sharad Sharma for his support and inspiration during the working on this project. The University of North Texas (UNT) (https://unity.com/)-supported free software of Unity 3D was employed to work on this project, accompanied by other programs such as Vizard (https://www.worldviz.com/products/vizard) and 3ds Max (https://www.autodesk.com/products/3ds-max/overview).

Everything that was written, textured, and 3D modeled that was used in the project was sourced from free resources like Unity Asset Store, Sketchfab, OpenGameArt, and GitHub projects. Everything else of the assets was used under proper license with proper credit given wherever it was required.

References

[1] T. Lin, "Fear in Virtual Reality (VR): Fear Elements, Coping Reactions, Immediate and Next-day Fright Responses Toward a Survival Horror Zombie Virtual Reality Game," *Computers in Human Behavior*, vol. 110, 2020, Art. no. 106383, doi: 10.1016/j.chb.2020.106383.

[2] S. Ulmer, M. Schäfer, T. Rohleder, and P. Klimant, "Gamification of Virtual Reality Assembly Training: Effects of a Combined Point and Level System on Motivation and Training Results," *Procedia CIRP*, vol. 107, pp. 1123–1128, 2022, doi: 10.1016/j.procir.2022.05.169.

[3] E. S. de Lima, B. M. C. Silva, and G. T. Galam, "Adaptive Virtual Reality Horror Games Based on Machine Learning and Player Modeling," *Entertainment Computing*, 2024, doi: 10.1016/j.entcom.2024.100536.

[4] D. Scorgie, Z. Feng, D. Paes, F. Parisi, T. W. Yiu, and R. Lovreglio, "Virtual Reality for Safety Training: A Systematic Literature Review and Meta-Analysis," *Safety Science*, vol. 154, 2022, Art. no. 105887, doi: 10.1016/j.ssci.2022.105887.

[5] C. Strannegård, N. Engsner, S. Ulfsbäcker, S. Andreasson, J. Endler, and A. Nordgren, "Survival Games for Humans and Machines," *Cognitive Systems Research*, vol. 83, 2024, Art. no. 101235, doi: 10.1016/j.cogsys.2024.101235.

[6] D. Herumurti, A. A. Yunanto, G. A. Senna, I. Kuswardayan, and S. Arifiani, "Development of First-Person Shooter Game with Survival Maze Based on Virtual Reality," in *Proc. IEEE Int. Conf.*, 2020, doi: 10.1109/ICASTE52004.2020.00045.

[7] S. Singh and A. Kaur, "Game Development using Unity Game Engine," in *2022 3rd International Conference on Computing, Analytics and Networks (ICAN)*, 2022, pp. 456–460, doi: 10.1109/ICAN55393.2022.10041654.

[8] A. Meliones and I. Plas, "Developing Video Games with Elementary Adaptive Artificial Intelligence in Unity: An Intelligent Systems Approach," in *2017 Intelligent Systems Conference (IntelliSys)*, London, UK, 2017, pp. 724–730, doi: 10.1109/IntelliSys.2017.8324313.

[9] Armanto, Hendrawan, et al. "Implementation and Impact of Virtual Reality on Survival Horror Games." *Inform : Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, vol. 6, no. 2, 28 July 2021, pp. 87–92, https://doi.org/10.25139/inform.v6i2.3943. Accessed 17 Dec. 2021.

[10] Pallavicini, Federica, et al. "Effectiveness of Virtual Reality Survival Horror Games for the Emotional Elicitation: Preliminary Insights Using Resident Evil 7: Biohazard." Universal Access in Human-Computer Interaction. Virtual, Augmented, and Intelligent Environments, 2018, pp. 87–101, https://doi.org/10.1007/978-3-319-92052-8_8.

[11] Arnold, Peter, et al. ""You Better Eat to Survive."" *Tangible and Embedded Interaction*, 18 Mar. 2018, https://doi.org/10.1145/3173225.3173238.

[12] Cecotti, Hubert, et al. "Countdown VR: A Serious Game in Virtual Reality to Develop Mental Computation Skills." *IEEE Transactions on Games*, 2024, pp. 1–12, https://doi.org/10.1109/tg.2024.3357452. Accessed 14 Feb. 2025.