Airport Security Simulation: A Multiplayer Virtual Reality Airport Training Simulator

Sincere Thanks to our Professor Dr. Sharad Sharma

1st Uma Abhishek Polakonda
dept. of Data Science
College of Information Science
University Of North Texas

2nd Vijay Venkatrao Goparaju
dept. of Data Science
College of Information Science
University Of North Texas

3rd Bala Phaneendra pothuri
dept. of Data Science
College of Information Science
University Of North Texas

***Abstract*—Airports are some of the most advanced, high-security buildings in the world. Training for airport security procedures is typically resource-demanding and operation-disruptive. This project proposes a multiplayer Virtual Reality (VR) simulation of an airport environment where users can role-play as passengers or security personnel. Built using Unreal Engine, the simulation emphasises authentic traveller processing processes such as boarding pass retrieval, baggage handling, and security screening. Security personnel are responsible for screening passengers and inspecting luggage. The project targets aviation students, airport management trainees, and VR developers, providing an entertaining and educational platform for airport learning through rich, interactive VR environments.**

***Index Terms*—Virtual Reality, Multiplayer Simulation, Airport Security, Unreal Engine, AI Behaviour, VR Interaction**

I. Introduction

The workings of an airport comprise many different workflows: passenger flow, baggage handling, security screening, and strong and effective surveillance. The very complexities that define these environments make them prime candidates for simulation training. Virtual Reality gives these airports the opportunity to simulate their complex workings without interfering with the real-life operations so that learning happens in an absolutely safe environment.

The aim of this project is to build an airport security system modelled as a fully-fledged simulation system capable of multiplayer VR support. The players can either act as passengers moving through the airport toward boarding, or even as security staff ensuring that the airport is secure. The world simulates the area of airport lounges, check-in counters, baggage scanning spots, and boarding gates. The environment is alive with the presence of users and AI characters.

The environment allows the user to dynamically interact with the environments, complete specific role-oriented tasks, and observe dynamic AI behaviors. Multiplayer functionality is incorporated to facilitate cooperative play among passengers and security personnel, thus fully immersing a user in a security process simulation.

A. Goals and Objectives

The objectives are:

- Establish a realistic, detailed 3D airport environment optimised for VR.
- Set up a multiplayer system offering varying perspectives of airport flow.
- Simulate realistic workflows for passengers and security personnel.
- Create AI-powered roaming passengers for environmental realism.
- Design intuitive VR interaction systems for object manipulation.

II. Modelling the VR Environment

Airport VR is designed with high fidelity, reflecting an airport in the real world. The model features terminals, check-in counters, and baggage handling. The powerful graphics capabilities of Unreal Engine enable detailed textures, realistic lighting, and shadow effects, making the simulation immersive and visually appealing. The design features are:

- **Environment:** This encompasses the city environment surrounding the Airport, the mountain and the taxis, roads.
- **Airport Environment:** This encompasses the airport model, baggage counters, security terminals, and waiting lounges.

Textures and shaders are used to generate realism. High-quality textures are utilised within the airport for tactile surfaces that include wood texture for floors, ceramic textures for walls, glass textures and so forth.

III. Application Usage and Interactions

The Security Guard Training VR app presents an immersive, first-person experience meant to mimic the duties of airport or high-security personnel security guards. Players perform a range of tasks, including screening suitcases on virtual X-ray machines, watching surveillance cameras, and conducting bag searches for possible threats. The training simulation presents a series of training scenarios in which the user must spot suspicious objects to provide protection.

The game presents tasks that reflect actual scenarios, e.g., dealing with emergency responses and managing high-traffic areas. Players are also tasked with reporting incidents and adhering to security protocols, so they are knowledgeable of the actions to take in case of security breaches or threats. Training modules become increasingly advanced, putting players in a position of having to think fast and make quick choices under pressure.

Directed at security staff, training facilities, and emergency response units, the VR app provides a controlled and safe platform for security procedures to be practised. Through its immersive and interactive interface, users are able to hone their skills in threat identification, surveillance tool utilisation, and responding to emergencies without the hazards of live training.

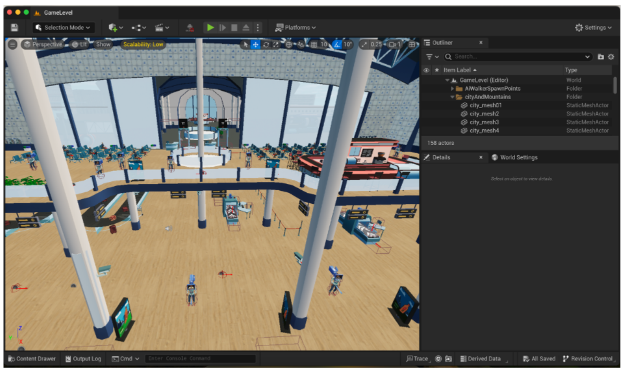

Fig. 1. Overall Layout of the Airport Simulation including Terminals, Security Checkpoints, and Waiting Areas.

IV. Related Works

Various games, tutorials, and literature have inspired this project. Some of the most important references are as follows:

A. Games

- **Airport Simulator 2019:** A game where players manage an airport, focusing on operations like aircraft management, baggage handling, and passenger services.
- **SimAirport:** A simulation management game where players design and operate airports while balancing growth and efficiency.
- **Flight Simulator 2020:** Although not strictly a management simulation, it provides high-fidelity airport environments useful for understanding airport design and operation.

B. Tutorials and Learning Resources

- **Unreal Engine VR Templates:** Tutorials on setting up VR environments within Unreal Engine to build interactive simulations.
- **Constructing an Airport Simulation:** Articles from platforms like GameDev.net, covering game mechanics, AI, and environment modeling for airport simulations.

C. Books

- **Learning Unreal Engine VR Development** by John P. Doran — A comprehensive guide to constructing VR simulations using Unreal Engine, providing insight into game mechanics and VR interaction design.
- **Game AI Pro** by Steven Rabin — Offers thorough knowledge on AI behaviour, including pathfinding and decision-making systems for NPCs within simulations.

To conclude, the literature reviewed emphasizes the importance of immersion, flexibility, motivation, and realism in VR simulations. Our project synthesizes these research outcomes to provide a substantial and interactive experience, laying the foundation for future improvement through player modeling and adaptive AI.

V. System Implementation Flow and Structure

The following diagram shows the sequential step-by-step process conducted during the development of the VR-based airport security simulation.

It starts with 3D modelling, exportation, and importation of assets into the Unreal Engine. It proceeds to scripting, adding behaviour, and integrating gameplay mechanics. User interaction with the VR client is then implemented, leading to the final phase of testing, debugging, and optimisation. This systematic step-by-step process ensures a coherent development workflow.

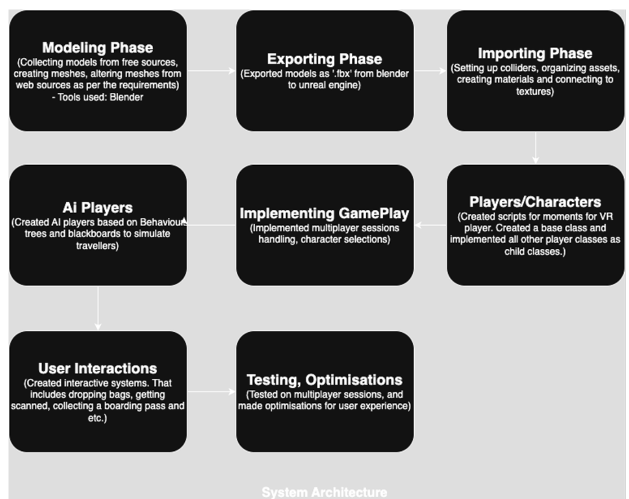

Fig. 2. System Architecture Diagram displaying Player Inputs, Multiplayer Server, AI Crowd, and VR Environment Interactions.

A. Modelling Phase

The visual environment was sourced and constructed by downloading airport models from Sketchfab and CGTrader. These included terminal buildings, waiting areas, kiosks, and baggage scanners. Models were post-processed in Blender to trim excess geometry, remap UV textures, and model missing elements such as conveyor belts and check-in counters.

After optimisation, models were exported as FBX files and imported into Unreal Engine 5, forming a functional airport layout for player navigation. Physically Based Rendering (PBR) materials were applied to accurately simulate common airport surfaces like metal, plastic, and fabric. Lighting setups created a bright, modern terminal atmosphere.

B. Game Mechanics

The core mechanics revolve around task delegation and real-time decision-making. Players act as security officers, overseeing tasks such as baggage scanning, passenger flow management, and adherence to safety protocols. Resource management includes handling airport staff and equipment while prioritising boarding, baggage loading, and emergency responses.

Time pressure and unexpected events, such as delayed flights or emergency incidents, are introduced to challenge players to adapt quickly and make decisions under pressure.

C. AI Behaviour

The simulation employs advanced AI systems governing non-playable characters (NPCs) such as security personnel, ground staff, passengers, and potential threat agents. NPCs perform behaviours such as queue management, crowd navigation, and dynamic decision-making.

Behaviour Trees and Blackboard variables simulate realistic actions. For instance, arriving passengers interact with boarding pass kiosks, drop baggage at assigned counters, and proceed through security checkpoints. If queues are long or terminals are busy, NPCs adjust their behaviours accordingly. In case of security alerts, AI agents respond based on their assigned roles.

D. Interactions

Players interact with the environment through VR controllers, mimicking actions such as operating doors, handling luggage, and monitoring passenger screening. Flight board updates, baggage management, and security status monitoring are handled interactively.

Haptic feedback provides physical sensations when interacting with objects like luggage or checkpoints, enhancing immersion. Players can scan and tag suspicious passengers or objects, initiate police responses for potential threats, and conduct direct searches for contraband items. These interactions emphasize attention to detail, quick reactions, and decision-making under realistic security simulation scenarios.

VI. Functionality Section

A. Vision

Obtaining visual realism was one of the key goals of the project. High-quality 3d assets were collected from Sketchfab and CGTrader to create a plausible airport environment. These assets were brought into Blender, where performance costs were minimised by optimising unnecessary geometry, UV mappings were fixed, and scales were normalised to real-world dimensions.

In Unreal Engine, each model was textured using Physically Based Rendering (PBR) materials to emulate surface properties like shiny floor tiles, reflective glass walls, metallic baggage scanners, and soft fabric seats. Dynamic lighting and baked global illumination were utilised to produce natural brightness and realistic shadows within the terminal. Other post-processing effects, such as ambient occlusion and bloom, were also used to add depth perception and material reflectiveness. This thoughtful visual arrangement creates for players an immersive, spatially correct airport that approximates real-world terminals.

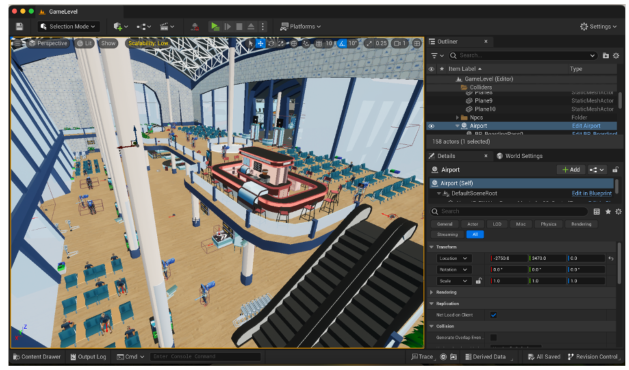

Fig. 3. Detailed Layout of Waiting Lounge showing Modeled Seating and Lighting Setup.

B. Sound

Sound design played an important role in creating player immersion and bringing the airport to life. A sound layering scheme was achieved by using spatialized 3D sound emitters. Ambient sounds like crowd hums, PA announcements, machinery conveyor sounds, and security scanner beeps were scattered around the terminal through Unreal's Audio Component system.

Every sound source was set up for volume reduction according to player distance and direction, providing a smooth auditory transition as the players transitioned between areas such as waiting lounges, security portals, and boarding gates. Furthermore, certain interaction sounds like the metal slide of doors or the conveyor belt movement were activated on specific player actions via Blueprints. The mixture of ambient and triggered sound realistically replicates the cacophonous environment of an actual airport terminal.

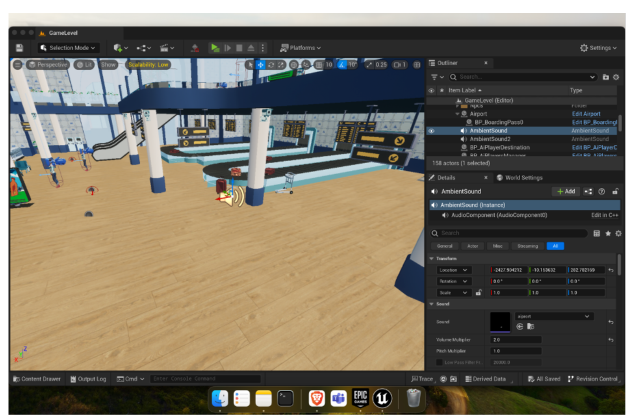

Fig. 4. Sound Emitter Placement throughout Terminal Zones.

C. Animation

Animations give life to both AI characters and environmental objects. AI travellers were rigged and animated through the use of motion-capture animations from Mixamo, such as walking, sitting, standing, and idling. To prevent repetitive motion, Blend Spaces were implemented in Unreal to transition between animations smoothly based on AI agent speed and movement states.

Environmental elements like sliding doors were made animated with Timeline nodes, where doors could open and close smoothly on proximity detection. The conveyor belt system was animated with constant AddForce physics applied to the baggage models, making bag movement realistic. A custom scanning animation was also made using Unreal's Animation Editor, in which security officers can realistically scan players before they board. These synchronised animations provide a dynamic, interactive environment in which players are always aware of motion and interaction.

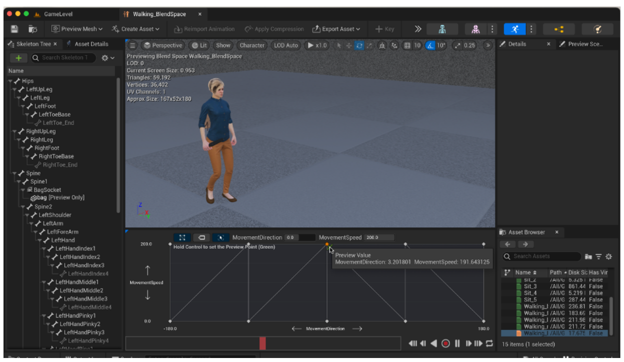

Fig. 5. AI Character Animation Sequence showing Traveller Approaching Security Gate.

D. Interactivity

Interactivity is the basis of the player's interaction with the simulation. A system of interaction was created within Unreal Engine through Blueprints and collision volumes. When a player is inside an assigned interaction area (such as in front of a boarding pass kiosk, baggage conveyor, or security checkpoint), an on-screen message pops up, enabling the player to click on an interaction button to have an action done.

Blueprint Event Dispatchers and Multicast Events were used to mirror such interactions throughout the network so that all the players can see object state changes (e.g., a door opening when a player walks through). Object-specific interactions are scripted independently: boarding passes are taken up and allocated terminals; bags are loaded on conveyors and pushed physically; body scanners trigger on when players walk through. This modular interaction system guarantees scalability when introducing future interactive elements.

E. Characters/Avatars

Players are represented by custom VR avatars that are distinguished by which role they chose: traveller or security officer.

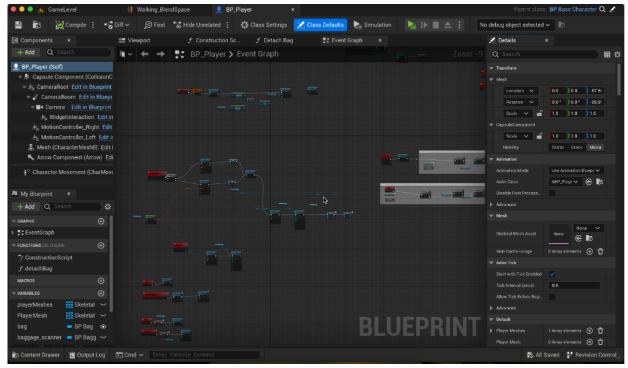

Fig. 6. Interaction Blueprint with Event Trigger and Replication Setup.

The VR player controller was altered to support VR hand presence, teleportation, and interaction capabilities tailored to role-specific tasks.

Along with player avatars, AI travellers fill the airport. AI agents have randomised appearances, behaviours, and movement patterns to mimic various crowds. With Skeletal Mesh Components and Animation Blueprints, AI travellers execute walking, sitting, waiting in lines, and idling activities, giving the impression of a crowded airport.

F. Sensors

Several forms of sensor systems were used to power in-game interactions and automate environment reactions:

- **Proximity Sensors** for doors and scanners.
- **Touch Sensors** for baggage drop points.
- **Time Sensors** for scanner delays.

All these sensors work together to form a responsive, interactive setting that responds naturally to player actions.

G. Player Controller

The VR Player Controller was created by expanding Unreal Engine's default VR template. Changes involved:

- **Teleportation Locomotion:** Allowing players to locomote easily throughout large airport zones.
- **Interaction Systems:** Seamlessly integrating controller input bindings into interacting with local objects.
- **Role-Specific Functionalities:** Passengers and security personnel achieve distinct actions (e.g., passengers board terminals, security patrols scan passengers).

The controller was also modified further for multiplayer support so that animation, object grasps, and state changes were imitated accurately amongst players.

H. AI Implementation

A dynamic management system for AI was created based on a bespoke Blueprint Crowd Manager. AI characters are spawned in random locations across the airport and given tasks such as walking to gates, waiting in chairs, or queuing at security checkpoints.

Unreal Engine's NavMesh system handled AI pathfinding. AI travellers move smoothly without collision by establishing proper navigation bounds and avoidance settings. AI states (e.g., Walking, Idling, Seated) are controlled by Behaviour Trees and Blackboard assets, adding realistic variability. Performance optimisations were achieved through dynamic despawning of AI agents outside the player's view distance.

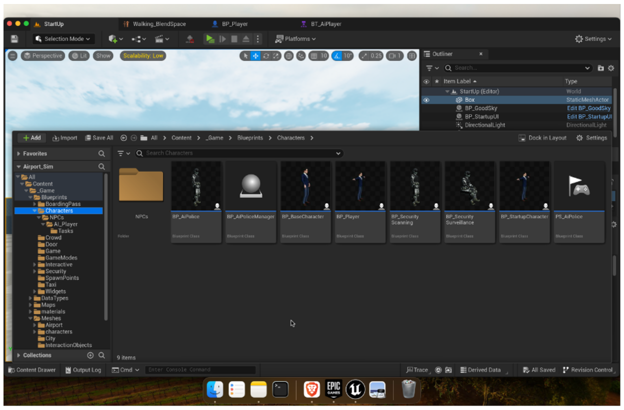

Fig. 7. NavMesh and AI Agent Pathing Layout in Terminal Environment.

I. Interface Elements

A minimalist and easy-to-use VR UI was devised to facilitate the multiplayer experience:

- **Session Menu:** Enabling players to create or join available multiplayer sessions.
- **Role Selection Interface:** Asking players to choose their role as a traveler or security.
- **Notification System:** Showing messages for successful boarding pass pick-up, security check clearance, and baggage placement confirmation.

All UI components were constructed with large font sizes, VR-friendly enough to ensure readability within headsets.

J. Multi-User Environment

Multiplayer capability was implemented with Unreal Engine's Online Subsystem. Players can create a session (server) or join and find existing sessions (clients). Fundamental interactive events such as opening doors, setting baggage, and scanning travellers are processed through Blueprint Replication and Multicast Events to synchronise state for all joined players.

Server control was managed with care to avoid collisions when several players engage with common objects at the same time. Testing of VR multiplayer was done with Oculus MetaQuest devices on the same network to ensure real-time performance and visibility of players.

VII. Conclusion

The Airport Security Simulation project does an excellent job of showing the potential for using Virtual Reality to simulate dynamic, realistic real-world environments for experiential learning, training, and education. By creating a realistic airport environment, adding interactive VR mechanics, dynamic AI crowds, and full multiplayer capabilities, the simulation is able to accurately and immersively replicate the experience of working in an airport.

This project emphasises the capability of VR not only to reproduce physical spaces but also to model procedural processes like baggage handling, traveller security screening, and gate boarding procedures. Involving players in role-playing exercises, the simulation promotes greater insight into the essential interactions between travellers and security agents within actual airport systems. Although the project effectively met its main objectives, difficulties were faced in optimising AI pathfinding performance, handling resource limitations in multiplayer replication, and ensuring smooth interaction among clients. These difficulties gave worthwhile experience in network synchronisation and VR performance optimisation.

Future development will involve deeper and more complex simulation through the addition of randomised security threats, adaptive AI behaviours reacting in real-time to crowd conditions, and adding the environment to include customs processing spaces and international flight terminals. Additional investigation of mobile VR platforms and cloud-based multiplayer services will improve accessibility and scalability.

In conclusion, Virtual Reality proved to be a perfect technology to simulate airport security operations, providing not just immersive involvement but also utility in training in a cost-effective and safe way.

References

[1] Airport 3D Model, Sketchfab. [Online]. Available: https://sketchfab.com/3d-models/airport-pack-total-ad40f27ecd5e492c8149cde6a5dd56d1

[2] Cityscape Model, CGTrader. [Online]. Available: https://www.cgtrader.com/free-3d-models/exterior/cityscape/low-poly-city-ab4d8e60-de2b-41d3-8669-632a4e240701

[3] Characters and Animations, Mixamo. [Online]. Available: https://www.mixamo.com/

[4] Backpack 3D Model, CGTrader. [Online]. Available: https://www.cgtrader.com/free-3d-models/various/various-models/backpack-a5b955d7-2fd5-4f51-99d1-7e4937c9cf9e

[5] Mountain Model, Sketchfab. [Online]. Available: https://sketchfab.com/3d-models/mountain-low-poly-cf8c27b8cc8344a78755505ee42e5a18

[6] Airport and City Sound Effects, Pixabay. [Online]. Available: https://pixabay.com/sound-effects/search/airport/

[7] J. P. Doran, *Learning Unreal Engine VR Development*. Packt Publishing, 2019.

[8] S. Rabin, *Game AI Pro: Collected Wisdom of Game AI Professionals*. CRC Press, 2013.

[9] J. Vora, S. Nair, A. K. Gramopadhye, A. T. Duchowski, B. J. Melloy, and B. Kanki, "Using virtual reality technology for aircraft visual inspection training: presence and comparison studies," *Applied Ergonomics*, vol. 33, no. 6, pp. 559–570, 2002.

[10] Constructing an Airport Simulation, GameDev.net. [Online]. Available: https://www.gamedev.net/articles/programming/general-and-gameplay-programming/constructing-an-airport-simulation-r3001/