

All the code can be found from <https://github.com/jiwonhaha/VECW1>.

1 Description

At the airport or the border of country, it is important to screen travellers before entering the country because there are many traveller having a misconduct or hiding a bad intention. Passport inspector needs to check their documents carefully to let the illegal travellers cause the problem to his country. This project is to simulate the situation at passport control of an international airport with Unity [1]. The purpose is to train the passport inspector to make the correct decision between permission to enter or rejection for travellers. The program will generate travellers with many randomized background to practice the procedure for the inspector.

1.1 Role Assignment

The simulation starts with participants selecting from three roles: Traveler, Inspector, and Supervisor. This choice determines the tasks and responsibilities each participant will perform.

1.2 Training

1.2.1 Traveller Role

The traveller begins the simulation by approaching a designated table to review passports and understand the purpose of each visit through an interactive UI. Holding a passport, the traveller then proceeds to the inspector for inspection, and answer a question about purpose of visiting.

1.2.2 Inspector Role

The inspector interacts with the traveller to inspect about their visit and scans the passport to get the traveler's detailed profile. Based on profile and answer from traveller, inspector should make a decision.

1.2.3 Supervisor Role

As the inspector scans the passport, the supervisor gains access to the system, where they can view the correct decision and its rationale. This allows the supervisor to assess the inspector's performance against a checklist, evaluating their rule understanding and decision-making skills.

1.3 Feedback

Following all rounds of simulation, participants return to the starting point. The outcomes and detailed analyses of the rounds are shared, allowing the supervisor to provide feedback and share insights. This phase fosters a reflective and collaborative learning environment, enhancing the overall training experience.

2 Design

Creating a realistic simulation environment was important to achieving the project’s educational objectives. The design concept aimed to closely mimic the real-world setting of an airport’s passport control area, focusing on the visual and interactivity to enhance the immersive training experience for participants.

2.1 Assets

To build our simulation environment, we obtained assets from two different ways:

1. **Unity Assets Store Resources:** The interior accessories of our assets are mostly imported from Unity Asset Store to create a realistic scene of passport control room and emphasis the detail. The usage is *Office Pack* [2] because it has the most similar items for our scene.
2. **Self-Designed Objects:** To supplement the assets obtained from the Unity Asset Store and to cater to specific design needs not covered by available resources, we turned to *Blender* for the creation of custom 3D objects. The room itself is the showcase where we design to mimic the airport scene. There are also other objects that are made by our group, for instance, inspector booth, monitor, passports, and passport scanner.

2.2 Final Design

2.3 Lobby

The lobby is a room where players choose their role first and sharing feedback after simulation. At the center, networking manager containing joining code is located, and the role assigning interface is placed at the front. The furniture inside the room is used to make it look as office of immigration officer.

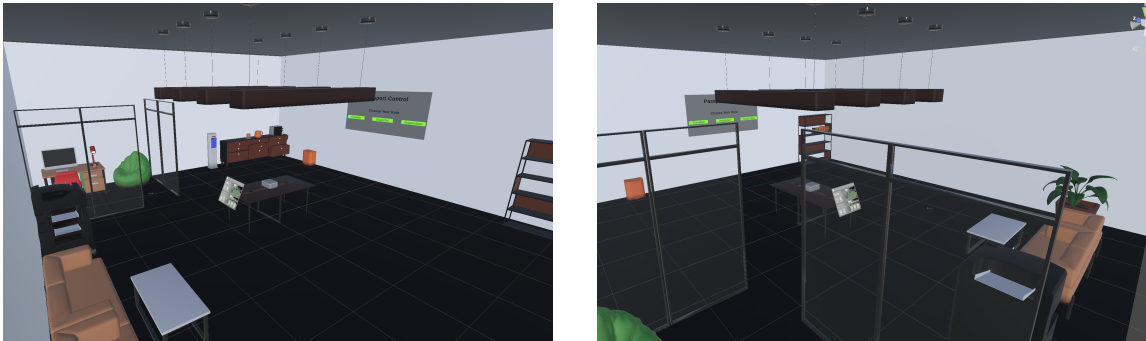


Figure 1: Lobby room

2.4 Main Room

The main room is created based on passport control of the UK airport. A hallway has 4 advertisements, 2 bots, and 1 table. 4 advertisement and view of outdoor is generated by *DALL-E* to make more realistic. Passing through the hallway, there are retractable barrier for guiding traveller's path, and 3 booths designed for inspector, supervisor, and bot player. Each room has its own desk and window. In inspector booth, there are a monitor, passport scanner, decision buttons and visa-free country list which is only visible to inspector, and as for supervisor booth, a monitor and decision buttons are set with instruction checklist which is only visible to supervisor.

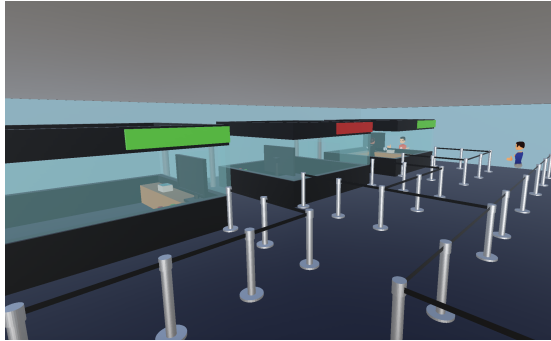


Figure 2: Gameplay Room (Hallway)



Figure 3: Gameplay Room (Booth Room)

3 Code

3.1 Gameplay

Initially, players select roles from inspector, supervisor, and traveller in the lobby, each with distinct tasks. Each player gets instruction for their role. The game starts when each player has chosen a unique role, teleporting them to specific locations with tailored instructions. Travellers receive a random passport and story via pop-up, while inspectors and supervisors are teleported to their booths. Travellers approach the inspector's booth with their passport, and the inspector scans it. This scan displays the traveller's information on both the inspector's and supervisor's monitors, but only the supervisor sees the correct trial answer. Inspectors decide on entry permission using buttons, based on passport details and response about purpose from traveller, and supervisors evaluate the decision of the inspector. After all rounds, the game returns players to the lobby, displaying round results, including passport information, inspector decisions, and supervisor evaluations, on the right wall.

3.2 Networking

The networking part uses an open source called *Ubiq* [3]. The process is exchanging the messages between players and server. One player will send a network message to the server, and others receive it and process as the function.

3.2.1 Passport

Passport is the object that will be used to interact most during the game because it is needed to be held by players and moves smoothly. The passport will send a message to server, which contain 2 variables:

- bool **isHolding** - to state whether it is held by player or not
- Vector3 **position** - the position of passport

When it is grabbed by player, *isHolding* is set to true and *isKinematic* to true to avoid rigid body collision, including sending its position to server, so others' can update their position. After the player releases it, *isHolding* and *isKinematic* is set to false, so it can interact with world physics.

3.2.2 Button

The results from buttons on inspector and supervisor desk are needed to be recorded for game system, so the button is implemented with networking object. The function is applied that the button can be pressed while *isInGame* is set to true. When the button is pressed, it will send a message to a server that the button is pressed, and cannot be pressed again in the same round. It will be reset when *isGameReset* set to true.

3.2.3 Game System

To handle multiplayer simulation system, the networking message is consist of 5 variables:

- int **totalOftraveller** - The total number of players choosing traveller role
- int **totalOfinspector** - The total number of players choosing inspector role
- int **totalOfsupervisor** - The total number of players choosing supervisor role
- List<int> **passportIndices** - The list of passport indices in the game.
- int **token** - The token specifying the main list of passport indices in the game.

Before the game start, there should be 3 players inside the room. The system will get the child objects of avatar manager. If there is less than 3 players, the existing players cannot choose a role. Once the player choose a role and the number of assigned player of that role is equal to maximum number, the button on screen is set to red color and other cannot choose. If *totalOftraveller*, *totalOfinspector* and *totalOfsupervisor* are equal to maximum numbers of the specific roles, the game will start.

At the start of the program, each player will have a list called *passportIndices*. To avoid duplicate passport of each round and record the indices for final evaluation, the list will randomly choose different passport indices for number of rounds. The challenging part is that every player must have the same passport indices. Otherwise, other players cannot see the passport moving. As it is mentioned above, *token* is used to specify the main list that will be used in the game. The program set traveller's *token* to 1, and others' to 0, so the main list is the one that belongs to travellers' and replace others' with the traveller's. The message will be sent when player choose a role, and when the game finishes every round.

4 Contribution

4.1 Plan & Designing

The initial step involved brainstorming and selecting the project topic. After deciding on creating a VR simulation for passport control training, we designed the traveller's stories and the logical cases that needed to be considered for the simulation. This foundational work laid the groundwork for the development step that followed.

4.2 Asset Creation & Integration

Using Unity Asset Store resources and custom designs from Blender, we built a realistic environment for the passport control simulation, including the interior of the passport control room and all necessary objects such as booths, monitors, and passports.

4.3 Networking & Functionality

After that, we focused on the networking aspect, ensuring smooth communication between players and the server using the *Ubig* [3] framework. Also, core gameplay functionality, implementing the interactions between travellers, inspectors, and supervisors as per the simulation's requirements is developed.

4.4 Testing & Debugging

After all implementation we were working on testing and debugging step, identifying and fixing issues to refine the simulation. Special focus was given to networked interactions and gameplay mechanics, ensuring that the simulation runs smoothly across multiple devices.

References

- [1] J. K. Haas, “A history of the unity game engine,” 2014.
- [2] nappin, “Office pack,” <https://assetstore.unity.com/packages/3d/props/interior/office-pack-free-258600#description>, 2023, unity Asset Store.
- [3] S. J. Friston, B. J. Congdon, D. Swapp, L. Izzouzi, K. Brandstätter, D. Archer, O. Olkkonen, F. J. Thiel, and A. Steed, “Ubiq: A system to build flexible social virtual reality experiences,” in *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3489849.3489871>