Dipartimento di Informatica, Bioingegneria,
Robotica e Ingegneria dei Sistemi

Virtual Reality

Report

Control Tower for Drones

Advisor:
Alice Maria Catalano
Youssef Mohsen Mahmoud Attia
Giacomo Lugano
Hussein Ahmed Fouad Hassan

UNIVERSITY OF GENOA, JANUARY 2023

Member list & Workload

No.	Full name	Student ID	Percentage of work
1	Giacomo Lugano	5400573	100%
2	Youssef Mohsen Mahmoud Attia	5157341	100%
3	Hussein Ahmed Fouad Hassan	5161612	100%
4	Alice Maria Catalano	5157341	100%

Contents

1 Introduction

To be sure that the user is able to sense also behind buildings the drones use a special material for outlining them thanks to *Render custom depth feature* activation [1]. In addition we used widgets implementing the *ray-casts* [2] and *LineTraceForObjects* [3] to keep the display of the information menu active when the user looks at the drones showing: drone name, position, orientation, battery level, mission number and drone camera view.
We encountered difficulties while working with the *blueprint* language and Unreal Engine, as well as ensuring that the drones remained in the user's field of view. Our solution was designed for a simulation, which presented additional challenges with respect to the real situation in which most of the processing task would be easier.

2 State of Art

Remote control towers, also known as digital or virtual control towers, are a technology that allows air traffic controllers to remotely monitor and control an airport's airspace and runways from a remote location. This technology has the potential to revolutionize the way that airports operate, as it can significantly reduce the cost of building and maintaining traditional control towers, while also increasing the efficiency and safety of air traffic control.
According to a blog post by dblue.it, remote control towers use a combination of sensors, cameras, and advanced computer systems to provide controllers with a real-time view of the airport and assist in the identification of potential conflicts and the suggestion of courses of action.

One of the main benefits of remote control towers is their ability to reduce the cost of air traffic control. Traditional control towers are expensive to build and maintain, and they require a large number of trained controllers to staff them. In contrast, remote control towers can be operated by a smaller number of controllers and do not require the same level of infrastructure. This can make it more cost-effective for airports to implement remote control towers, particularly at smaller airports or at airports located in remote or challenging environments.

There are a number of remote control towers currently in operation around the world, including in Sweden, the United Kingdom, and the United States. These towers have been shown to be effective at providing controllers with the information and tools they need to safely and efficiently manage air traffic and have the potential to be deployed at a wide range of airports in the future.

In conclusion, the use of remote control towers is a promising development in the field of air traffic control. By allowing controllers to remotely monitor and control an airport's airspace and runways, these towers have the potential to significantly improve the efficiency and safety of air traffic control, while also reducing the cost of building and maintaining traditional control towers.

3 Tools

As software tools we used just Unreal Engine 5.0.3
Because of the GitHub limitation of file size in the repository there will just be the link to the One Drive directory, Here

4 Description

The project involves simulating a scenario in which four drones are deployed in a city-like environment. The environment is perceived through the perspective of a character that represents the user, equipped with a laser that detect and provide information on the nearby drones when they collide.
To set up the simulation, the first step was to create a realistic environment. To do this, the team searched for suitable objects in the UE5 marketplace, downloaded them, and assembled them together to create a city-like setting. The team also created models for the drones and the user character.
With the environment set up, the team designed three hypothetical missions for the drones to complete:

1. Building scan: In this mission, the drone approaches a building and stops for a few seconds in a position to perform a photogrammetry of the building. It then moves on to the next position.

2. Field scan: This mission simulates a search and rescue operation, with the drone systematically flying over a field looking for any signs of distress. In this case, the field is free of any obstacles, so the mission will appear more like a standard field scan.

3. Street scan: In the final mission, the drone follows a predetermined path along the center-line of a street, covering the entire map as it goes.

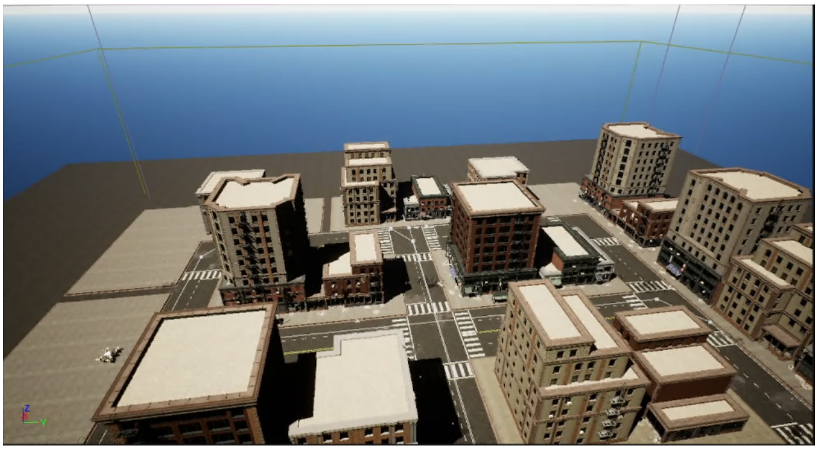
The initial goal of the project was to retrieve status information from the drones. To do this, the team had to consider how the user would perceive the drones and the limitations of using a laser for detection. It was determined that simply relying on sight would not be precise enough and could lead to confusion when multiple drones were present in the same view. Therefore, the team decided to use a laser as a more reliable means of detecting the drones. However, the team also recognized that the drones might not always be visible to the user, such as when they are behind buildings. To address this issue, the drones were equipped with a special material that allows them to outline their shape, making them visible to the user and easier to target with the laser. This is shown in the results chapter (Figure [insert figure number]).
Overall, the team carefully considered the various factors that would impact the user's ability to perceive and interact with the drones and took steps to ensure that the drones could be detected and identified accurately and efficiently.
Once the initial setup of the simulation was complete, the next step was to display the drones' information to the user. To do this, the team used widgets as tools to present the information on screen. These widgets worked in conjunction with blueprint code and other UE5 tools to retrieve and display the desired data.
For example, the team used a raycast to detect when the user's laser collided with a drone. When a collision occurred, the raycast would cast the hit target along with the IDs of all the drones. If the hit drone's ID matched one of the IDs that had been cast, the cast would succeed. Based on this information, the team set a variable called `drone_seen` to indicate that a drone had been detected. In the blueprint code on the level dedicated to this task, the team had a reference to the character and checked the value of the `drone_seen` variable. If the variable was set to true, the team set the visibility of the widget (i.e., the menu that displayed the relative drone's information) to true. Within the widget, the team included a function that took the input of the relevant variables and displayed them as text. For example, to display the drone's position, the team called an object

referenced to the specific drone ID and used the `get_location` function to retrieve the position data. The same was done for the orientation (using the `get_rotation` function) and for the battery level and current mission. In addition to displaying the drones' information in a menu, the team also wanted to show the drones' camera views on screen. To achieve this, they used a set of variables and specialized components. Specifically, we used a 2D capture component instead of a traditional camera to capture the image of what the drone is seeing. This captured image is then projected onto a material as a texture, which is in turn displayed on the widget.

5 Results

As a result of the project, several images were captured showing the city-like environment and the behavior of the drones.

Figure 1: Environment

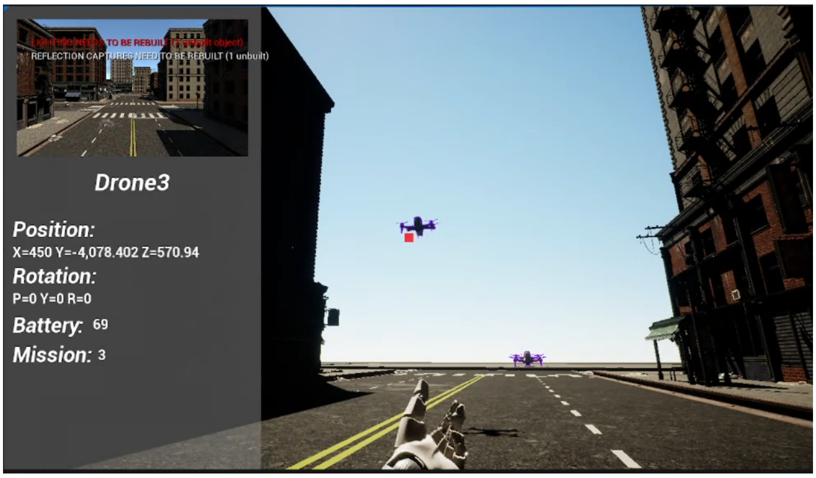


The attached images show the environment itself, with a complex of buildings visible on the right side of the picture. One drone is also visible in the left side of the scene. The field that was scanned during the second mission is located just behind this view.

Figure 2: Drone 1, display)

The attached image shows one of the drones' menus, displaying all the relevant information about the drone. Moreover the outline property of the robot material is shown, highlighting the possibility to check on its position through buildings. The menu also includes the camera view from the perspective of the drone, allowing the user to see what the drone is seeing in real time. This allows the user to easily monitor the drones and understand their behavior and capabilities within the simulation.

Figure 3: Drone 3, display



6 Conclusions

In conclusion, our team has successfully achieved all the objectives that we set out in this project. By creating a realistic simulation in the Unreal Engine 5.0.3 and designing missions for the drones to complete, we were able to simulate the behavior of autonomous drones in a city-like environment. We also implemented a system for the user to perceive and interact with the drones, using a laser and special material on the drones to make them visible and easier to detect. We used widgets and blueprint code to retrieve and display information about the drones. Overall, the simulation is a functional and realistic representation of autonomous drones interacting with their environment and with a user. Despite encountering some challenges along the way, we were able to overcome them and create a successful simulation. While we had many ideas for improvements and additional features like: user driving, having more than one drone camera appearing and other interaction features.

7 References

[1] Render Custom Depth

[2] raycasts

[3] LineTraceForObjects