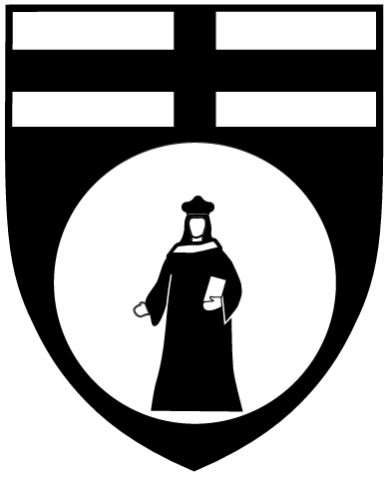


Dibris Dipartimento di Informatica, Bioingegneria,
Robotica e Ingegneria dei Sistemi

Virtual Reality class

Report

AI-based smart-ambulance

Advisor: Gianni Viardo Vercelli

UNIVERSITY OF GENOA, FEBRUARY 2024

Member list & Workload

No.	Full name	Student ID	Percentage of work
1	Isabel Cebollada Gracia	s5715592	65%
2	Abdelouadoud Guelmami	s5467288	13%
3	Baris Aker	s5430437	6%
4	Chloé Bohler	s6528470	16%

Contents

1 Introduction

In the ever-evolving landscape of technology and innovation, the realm of video game development serves as both a playground for creativity and a laboratory for practical applications. Within this sphere, Unreal Engine stands as a cornerstone, providing developers with powerful tools to craft immersive experiences that push the boundaries of what is possible. In this project, we delve into the intersection of gaming and real-world utility by focusing on the implementation of a medical aerial drone's movement within the cityscape of Genova, able to reach points through air that terrain vehicles can not reach due to external conditions.

In recent years the use of drones in various industries have appeared, offering unparalleled efficiency and flexibility in tasks ranging from surveillance to delivery. One particularly promising application lies in the realm of medical assistance, where drones have the potential to revolutionize the landscape of emergency response. The concept of using drones to transport medical supplies, particularly in scenarios where conventional ground transportation may be hindered by terrain or traffic, has garnered significant attention.

The objective of our project is to simulate the deployment of a medical aerial drone alongside an ambulance within the streets of Genova, Italy. Our focus is on seamlessly integrating the drone's movement with the ambulance's route, navigating through the bustling urban environment to reach remote mountainous areas where traditional vehicles may struggle to access. By leveraging the capabilities of Unreal Engine, we aim to create a realistic and immersive experience that not only showcases the technical prowess of game development but also highlights the potential impact of such technology on real-world scenarios.

By harnessing the capabilities of Unreal Engine to simulate real-world scenarios, we endeavor to inspire not only gamers but also policymakers, healthcare professionals, and technologists to explore the possibilities of utilizing drones for the betterment of society. As we embark on this journey, we invite you to join us in envisioning a future where innovation in gaming transcends virtual boundaries to make tangible contributions to the world we live in.

2 State of Art

In recent years, the integration of drones into various industries has accelerated, driven by advancements in technology and the demand for efficient solutions. Drones have emerged as invaluable tools in fields such as surveillance, delivery, and now, medical assistance. Research has highlighted their potential to revolutionize emergency response systems, particularly in scenarios where traditional ground transportation is hindered. Utilizing open data and advanced simulation technologies, projects like ours aim to explore the feasibility and impact of incorporating drones into real-world medical scenarios. Publications such as [3] have underscored the significance of this approach, emphasizing the need for innovative solutions in emergency medical services. Additionally, articles like [4] have provided platforms for discourse on the practical implementation and ethical considerations surrounding drone usage in healthcare. Our project seeks to contribute to this body of knowledge by simulating the seamless integration of medical drones into urban emergency response systems, with the ultimate goal of improving patient outcomes and resource allocation.

2.1 Drones on difficult areas

Drones have multiple advantages over terrain vehicles allowing to reach areas that can not be accessible with ground vehicles due to environmental conditions or external factors:

- Drones are able to fly, allowing to fly over obstacles that may be a problem for a terrain vehicle. This factor allows drones to reach areas with more probability than ground vehicles, since they can avoid in more ways the obstacles.
- Related with the previous one, since drones can fly they have more degrees of freedom, so the can move in 3 dimensions while terrain vehicles only on 2, and they are also constrained by the terrain conditions.

Of course, there are also disadvantages that drones encounter against the terrain vehicles, since other factors must be taken into account as the autonomous life or weather conditions.

3 Tools

3.1 Software Used

Below is the list of all the tools used for the development of the project:

- **Unreal Engine v. 5.2.1**
  It is a video game development tool from the video game and software development company Epic Games. With this tool, developers have the ability to build a simulation, edit videos or sound, and render animations.
- **Cesium for unreal (with GoogleMap API)**
  It provides a high accuracy 3D geospatial ecosystem to Unreal Engine. By using Google Maps API, it provides with 3D Tiles and cloud-based real-world content from Cesium ion to Unreal Engine.
- **D-flight website**
  This tools allows to have knowledge about the aerial restrictions for the drones, such as restricted zones or restricted heights to flight.
- **Colosseum latest version**
  It is a simulator for robotic, autonomous systems, specifically drone testing, built on Unreal Engine. Is open-source and cross platform, and is a fork of AirSim, that Microsoft shutdown on 2022 and allows maintenance and support for newer versions of Unreal Engine. It provides a high quality simulation environment that simulates with high fidelity all the dynamics of aerial vehicles.
- **Visual Studio v.2022**
  It is an IDE for creating applications on Windows, Android or IOS among others.
- **Python v.3.12.1**
  Python is a high level object oriented programming language that allows to create codes for different purposes.
- **TCP Sockets**
  It is a software structure that allows communication through the network between a client and a server.
- **GitHub**
  It is an online software development platform used for storing projects and allow the collaboration on them.
- **Overleaf (Latex)**
  It is an online LaTeX editor that allows to write documents in collaboration with others.

3.2 GitHub repository

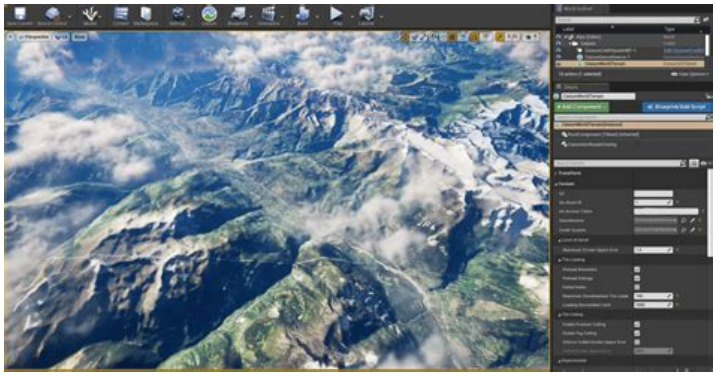
Find the link to our project here.

4 Description

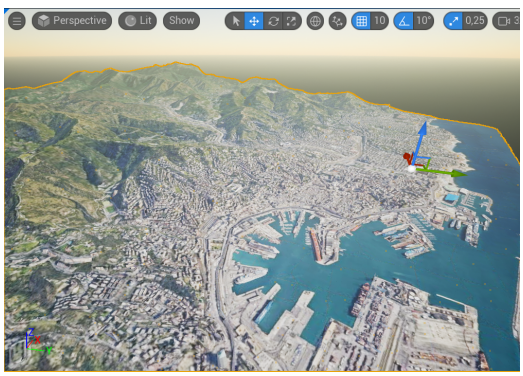
4.1 Genova Map

First of all, we need to implement an environment for our game. The desired idea is to recreate the Genoa map, in the goal to test and simulate our drone in a virtual environment where the drone must navigate.
To do so, we have started to learn more about cesium's capability. Cesium is a geospatial 3D mapping platform that enables the visualization of dynamic, high-resolution, and real-world geospatial data in a web browser. Infact, Cesium provides a robust set of APIs and tools for developers to create interactive and immersive geospatial applications.

In order use Cesium, we must create an account on this platform online. Furthermore, we have to download Cesium in Epic Game's library and add it to our Unreal Engine project. The next level is to open the project and to add the Cesium plugin in the list of games plugins (you must restart Unreal each time you add a plugins). Following, we have to connect our account to the Cesium plugin and create a new empty level to be able to use the "Georeference" panel of Cesium.
Thanks to this platform and to the "Tileset" module, we compute the google map API in order to use this information to provide an high-resolution entire world, as you can see in the following figure :



((a)) Entire world in Unreal Engine



((b)) Genova Map

However, in our project the goal is to have only the map of Genova, so we need to cut the map in order to keep only the Genova part. In the idea to product an game more efficiently and less ressourceful. To do so, we can use the"CartographicPolygon" module from Cesium and select the desired space with reverse the keeping selection.

Thanks to d-flight website, we are able to know where we have the authorisation to fly with a drone in Italy.

So, we can check at the mandatory flight zone :

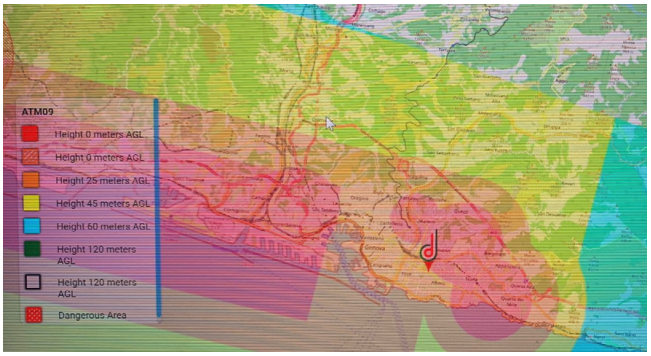


Figure 2: Genova map for fly authorisation

By using d-flight website we were capable to understand the in which zones of Genova how many we can fly with the drone. BY implementing this forbidden flight part in our project, we can create some invisible boundaries into space forbidden for drone flight. It provides to not fly in the restricted zones and also provides to fly according to the state regulations. D-flight map shows the areas by separating them with different colors to inform the user about the flight regulations in the city. Red zones are forbidden to fly by getting the zones' colour lighter means that the zones are flyable and when the color gets lighter the flyable height gets higher and higher. The zones that are near the airport and the hospital areas are forbidden to fly depending on to the regulations. (Invisible in game mode)

4.2 Drone implementation

In order to implement the drone inside Unreal engine 5.2. We will use Colloseum Github located in the following link: `https://github.com/CodexLabsLLC/Colosseum/tree/main/PythonClient/multirotor`
First thing we need to do is to install Visual Studio 2022. We should make sure To install the following requirements:

- Desktop Development with C++
- Windows 10 SDK (Or 11 last version)

We need to compile from the repository the Colosseum file. To so it is preferable to use **Developer Command Prompt for VS 2022**. Then we build the whole Colosseum with command **build.cmd**. In case of successfully building. Now we try to integrate the the drone in our Genova map. To achieve this it is necessary to paste the folder **Plugins** to the location of our Unreal Engine folder (Usually found in Documents). Also create a **C++ Class** inside the Unreal.
we should take into account that We need to modify two files before moving forwards. We need to add two line codes to **Map** and **DefaultConfig** which can be found in **Config** accordingly to `https://microsoft.github.io/AirSim/unreal_custenv/`
Last step to do is to run Both Map file and Visual studio. It is necessary to run **DebugGame Editor**. Now

finaly step choose in parameters of **World Settings** in **Override** choose **Airsim Game Mode**. Then press **Run** .
After these steps you should be able to see the Drone in the Unreal engine world. In the next parts we will explain how this drone can be controlled to do the requested task.

4.3 Ambulance & Drone Control

The task of the drone in this project is to arrive to places that the ambulance can not reach due to different situations, such as traffic or other obstacles that do not allow it to move.

4.3.1 Objects creation

For the development of this, there are two actors that take place on the Genova map: the ambulance and a fallen tree, that blocks the path for the ambulance. In this way, the ambulance will be stuck at some point so the drone will continue with the task.

To achieve this, it has been necessary create two new materials, since there are not available for free or by default ambulances and fallen trees in the Marketplace of Unreal Engine. To do this, first it has been necessary to create a material for both objects, adding the roughness, color base, metallic and normal according to each object (Figure 3).

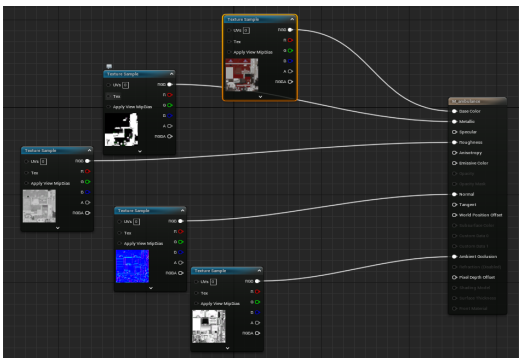


Figure 3: Ambulance material creation.

Once the material have been done, the static mesh has been imported to Unreal Engine, obtained from Sketchfab. Once the static mesh and the material are ready on Unreal, the only missing part has been to select to correct material to each static mesh. In this way, both objects have been ready to be used on the platform (Figure 4).

Figure 4: Ambulance object ready to be used.

4.3.2 Ambulance Movement

Once the objects are ready to be placed on the environment, it has also been necessary to create the movement of the ambulance. This task has been obtained by the use of blueprints, one for the ambulance and other for the path to be traveled. In the path blueprint, it has only been necessary to define a spline, while in the actor blueprint, some actions such as timeline, lerp, orientation, location and transforms have been used according to variables created in order to follow the spline in the correct time (Figure 5).

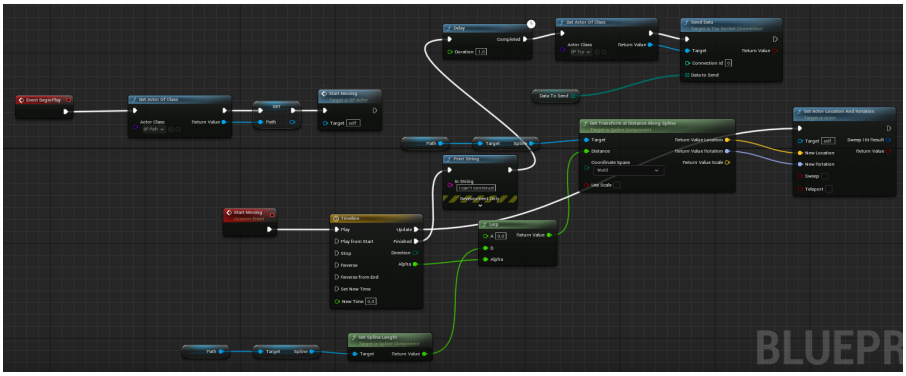


Figure 5: Ambulance Blueprint.

4.3.3 Connection between Ambulance and Drone

Finally, it is necessary to establish a connection between the drone and the ambulance in order to have a communication that indicates the drone when it has to start and where it has to go, as well as communicate anything if necessary. For the communication, sockets have been used under the TCP protocol. Both the drone and the ambulance connect to the same IP address under the same port in order to establish the connection. The drone acts as a client while the ambulance is the server that creates the connection. In this way, when the ambulance gets stuck because of the fallen tree, it immediately prints a message in the screen to let the user see that is blocked, and sends a message indicating the drone that has to continue the travel.

The development of the socket on the ambulance has been done though Blueprints. One blueprint has been

created to initialize the connection. By doing a blueprint cast, it has been possible to send the data to the drone when the ambulance gets stuck because of the fallen tree (Figure 5).

4.3.4 Drone Movement

Python has been used to use the Airsim API for the controlling of the drone. A distance detector sensor has been used to detect obstacles. In order to integrate it in the drone, the settings.json has to be modified in order to add this sensor (Figure 6). Once added, and by using the Airsim API, the distance to the closest obstacle in front of the drone can be obtained. In order to go to a known path, the Python script has been provided with known locations of mountain areas closed to Genova according to the limits in the flight heights, so the drone can follow a path while controlling the possible dynamic obstacles.

Figure 6: Drone with distance sensor.

5 Results

In this project, with the use of Collosseum –the new branch for newest Unreal Engine versions— and Unreal Engine 5, we have been able to achieve the desired behavior of a drone taking the role of an ambulance when this is stuck in some environments.

Due to the communication between the ambulance and the drone, it is possible to let the drone know when the ambulance got stuck in some place —as could be as shown in the project, because of a fallen try— and let it know where the goal was. After the drone receive this information and acknowledging the ambulance, it is able to fly through the environment until reaching the final destination, always taking into account the limitations of the flight heights. If the destination is not know by the drone —if it is not provided on the dataset— the drone will not fly and will notice the ambulance. By including more destinations in the dataset, the drone will be able to flight to more places not easy reachable by a terrain vehicle.

This project can be very beneficial for real applications, even more in environments as the city of Genova where there is uneven terrain that depends on very few bridges and roads to arrive to certain points, specially mountain villages, that can not be accessible due to different external conditions for a wheeled vehicle but still reachable for a flying vehicle as a drone, making possible to arrive to provide the needed medicines in the shortest time possible.

6 Conclusions

In conclusion, our project embarked on the ambitious task of implementing a virtual environment within Unreal Engine to simulate the movement of a medical aerial drone within the cityscape of Genova. To achieve this, we first focused on creating an accurate representation of the Genoa map using Cesium, a geospatial 3D mapping platform, seamlessly integrated with Unreal Engine.

Through meticulous research and implementation efforts, we successfully harnessed Cesium's capabilities to visualize dynamic and high-resolution geospatial data within our virtual environment. By leveraging Cesium's Tileset module, we tailored our map to focus solely on the Genova region, optimizing efficiency and resource utilization for our game.

Furthermore, our project delved into the implementation of the drone within Unreal Engine using the Colosseum framework. This involved intricate steps, including the installation of Visual Studio, compilation of the Colosseum files, and integration of the drone into our Genova map environment. Through diligent coding and configuration adjustments, we achieved the seamless integration of the drone into our Unreal Engine project.

The core objective of our project was to simulate scenarios where the drone aids in emergency medical response by reaching locations inaccessible to ambulances, such as areas obstructed by fallen trees or heavy traffic. Through the use of blueprints, materials, and communication protocols, we successfully orchestrated interactions between the drone, ambulance, and environmental obstacles within our virtual world.

Moreover, as technology continues to evolve, there is immense potential for the development of virtual reality (VR) and augmented reality (AR) applications in emergency medical training. By immersing trainees in realistic and interactive environments, VR and AR can provide invaluable hands-on experience and feedback, ultimately improving preparedness and response capabilities.

In summary, our project not only demonstrated the technical feasibility of integrating drones into emergency medical services simulations but also showcased the interdisciplinary collaboration between gaming, geospatial technology, and healthcare domains. Moving forward, our work paves the way for further advancements in utilizing virtual environments for training and optimizing emergency response procedures, ultimately contributing to the enhancement of healthcare delivery in real-world scenarios.

References

[1] Cesium for unreal.

[2] Colosseum for unreal.

[3] d flight. First flights with a drone to transport blood for analysis.

[4] UAvioni. What is the role of a medical drone in healthcare?