TOULOUSE INP N7

Long Project

Modifying the trajectory of a drone following a reconfiguration of the airspace.

2022-2023

Supervised by : Mr. Daniel ZAPATA

Bastien GERMAN, Chouaib LAAOUINA, Guohao DAI, Maha DRISSI-EL BOUZAIDI,
Mohammed-Amine JAAFARI, Othmane CHAOUCHAOU

9. mars 2023

Innhold

Abstract

In this report we present the project developed to explore the integration of drones into various industries and highlights the importance of safe and effective airspace management. The potential risks associated with drone usage, such as collisions and unauthorized surveillance, necessitate the development of technical solutions for drone management. This project presents a practical example of altering a drone's course in response to changes in airspace, utilizing the Leaflet JavaScript framework for real-time mapping and visualization, and the A* algorithm for determining the optimal trajectory.Potential improvements to the system include multiple drone processing, alternative algorithms, a notification system for human involvement, and 3D visualization for altitude representation. Ultimately, the success of drone technology and its widespread adoption will depend on the continued advancement of drone technology and the implementation of innovative solutions like the one presented in this project.

1 Introduction

Over the past few years, drone use has skyrocketed over the globe, and a productive infrastructure has been developed to support their uses. Drones are being used in a variety of industries, including aerial photography, agriculture, logistics, and even disaster management. Drone use is anticipated to increase further in the future with more autonomous capabilities and sophisticated activities.

Drones must be integrated into the airspace safely, effectively, and without compromising privacy as they become more commonplace. The idea of "U-space" has been created to address these issues. U-space involves a set of digital and automated functions and processes in defined airspace, aimed at ensuring the safe and efficient integration of drones, in other words, a U-space is like an air traffic control system for drones, allowing them to operate safely and seamlessly in the airspace.

A number of elements that improve the safety of drone operations will be included in the proposed system. For example, it will be able to follow and keep an eye on drone activities in real-time, which will assist to avoid collisions with other planes or objects.

The aim of our project is to develop a tool that allows for modifying the trajectory of a drone following a reconfiguration of the airspace. To do this, we had to develop an application that presents an airspace, simulates the movement of a drone in flight, simulates a reconfiguration of the airspace, and then modifies the trajectory either automatically or manually by the user.

In this report, we describe the management parts of our project, particularly how the six of us worked together and how we organized ourselves to ensure the smooth progress of the project, and we also describe the technical parts of this project that involves the design of a drone path configuration system that is intended to operate within the U-space framework. The technology tries to address some of the issues with safety that come with integrating drones into airspace,especially the modification of the trajectory of a drone following a reconfiguration of the airspace when imposing constraints on the trajectory initially planned.

2 State of the Art

In 2017, U-space was defined as a set of services and procedures in order to allow for safe, efficient and secure access to the airspace for drones.

While this set of services has been largely defined in its properties and functions, its implementation has been left to be taken care of by the EU member states.

U-space Services Implementation Monitoring reports have been released every couple years since 2018, and they show that even though no implementation has been launched yet, there is an increase in the number of stakeholders preparing to implement U-Space Services. Services provided for the U-space are divided in four categories : U1, U2, U3 and U4, also categorized as phases.

The report shows charts of the readiness for each aspect of the services, and indicates that no country in Europe as yet deployed any solution, but that many are planning and developing a solution, and that all countries of the Single European Sky (SES) are expected to be working on a functioning framework between 2021 and 2025.

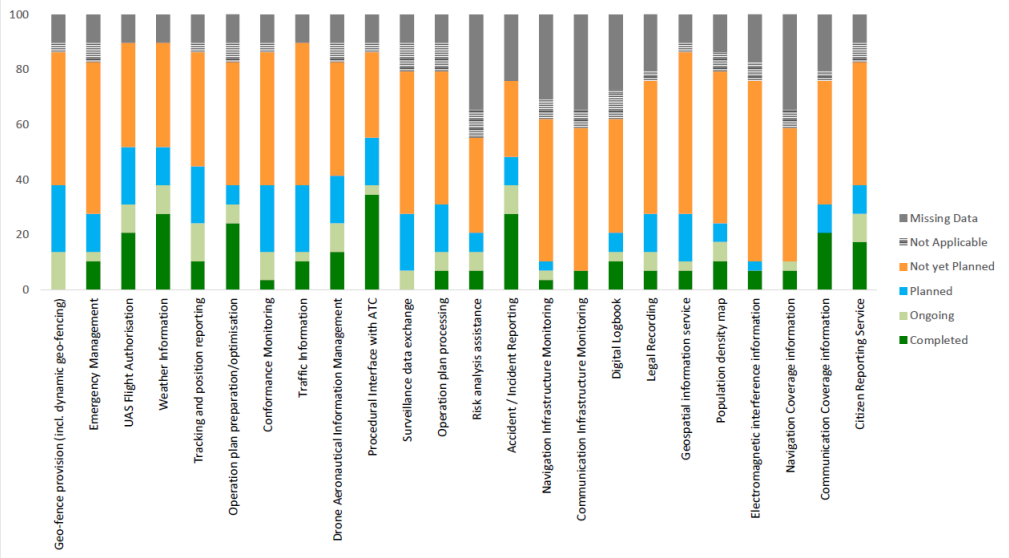

Figure 1: Graph of the readiness of implementation for U2 services

U1 represents the initial and basic services of U-space and U4 represents the complete implementation of all U-space services, and we have worked on implementing some of the U2 services concerning flight planning and drone operation management.

In France, l'Armée de l'Air is responsible for implementing the U-space services, and in most countries, governmental entities are responsible for it, while in some others, private organizations have been hired to work on this project. Since none of the projects have reached completion, and since the U-space project involves governmental entities, there were no source codes we could be inspired by, we had to base our work on the sole description of the services.

3 Methodology

In any project, a good approach involves, among other things, knowing how to search for information, take notes, manage time and activities effectively, and work harmoniously and efficiently in a team. In this regard, a development cycle and task planning have been specified to successfully complete this project.

3.1 Development cycle

Our work progressed through successive stages combining learning and practical application on parallel tasks. During the advancement of our project, we opted for the SCRUM Framework as an agile development methodology. To do so, we used **Trello** for project management.

Trello is a web-based project management and collaboration tool that allows users to create and manage boards, lists, and cards to organize and prioritize tasks and projects.

In order to properly manage our work, we first created the backlog, which contains the different major tasks of the project, starting with the aspects on which we were all to work together, namely: feasibility study, bibliographic study, choice of work tools, and then preparation of the work environment.

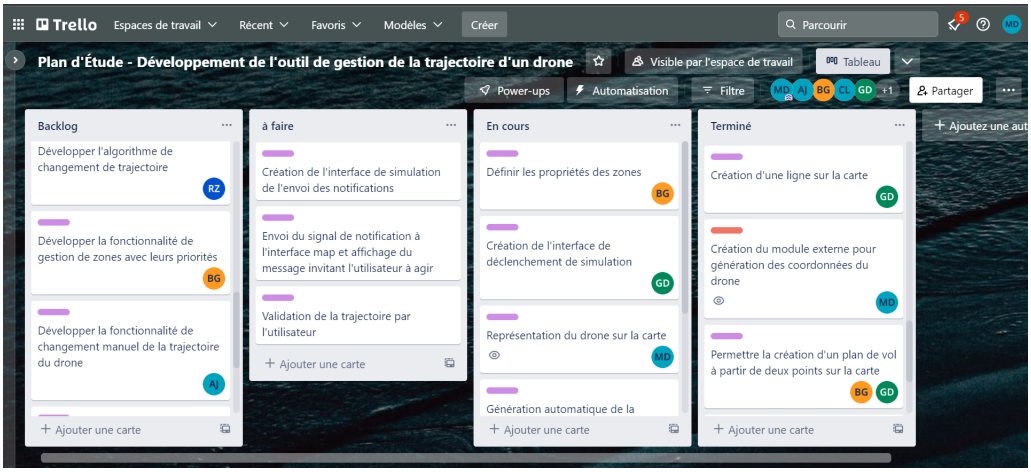

Figure 2: Trello project managment progress

Once the work environment was ready, we proceeded with task paralleling to optimize the time allocated to the project.

The aim of our project is to develop a tool that allows for modifying the trajectory of a drone following a reconfiguration of the airspace. To do this, we had to develop an application that presents an airspace, simulates the movement of a drone in flight, simulates a reconfiguration of the airspace, and then modifies the trajectory either automatically or manually by the user.

3.2 Work Organization

To carry out our work, and after choosing the right tools to work with, we began by dividing the project into several parallel tasks, which we then shared among the group members. Fur-

thermore, team meetings were held on a weekly basis and whenever it is needed, in order to be able to help each other solve problems and ensure that the work was progressing smoothly.

The main tasks described are as follows:

- Development of the basic interface and integration of the geographical map.

  This interface is the access point to the features offered by the developed tool. It should provide the user with a representation of the airspace and the different interfaces allowing them to manipulate it.

- Configuration of the airspace and creation of the drone's flight plans.

  This part consists of allowing the user to draw zones on the airspace and assign them a property that gives an idea of the availability of the zone. When a zone is unavailable, an algorithm is developed to provide a route that avoids flying over that zone. The development of the said algorithm is a whole different task mentioned in the next point. It should also allow the creation of a flight plan, represented by a route on the said space, corresponding to a drone with a specific identifier.

- Development of the algorithm that allows for the modification of the drone's trajectory following a reconfiguration of the airspace.

  This means that if at a certain moment, a zone that is being flown over by the drone becomes restricted or unavailable, the user must have the possibility to generate another path to avoid flying over the concerned zone using our tool. This new path is generated with the developed algorithm, which will be detailed in the following sections. Furthermore, it can be used for creating the initial path of the flight plan. For this purpose, the user should provide a starting and ending points and let the program define the path depending on the existing configuration of the airspace.

- Development of the external simulation module which simulates a drone in flight and generates its real-time coordinates.

  This module involves taking as input the flight plan of the drone in question, extracting the corresponding route, and generating coordinates that follow the path of the route. Then, these coordinates are sent in real-time to the user interface and represented by a drone on the map.

- Development of the functionality that allows for the manual modification of the drone's trajectory.

  This functionality will allow the user to redefine a new path to the drone themselves in case of a reconfiguration of the airspace if the solution proposed by the tool does not suit them.

3.3 Development Tools

After having a global idea of the project and the main functionalities to develop, the next step was to choose the tools to work with, and our choice was for the following languages and libraries:

3.3.1 Python

Python is a versatile programming language that can be used to develop a wide variety of applications. Additionally, the needs of our application and the time devoted to it led us to choose Python for the following reasons:

- Rapid development: Python is easy to learn and use, which can speed up the application development process.
- Large standard library: Python has a vast standard library that facilitates application development.
- Multi-platform compatibility: Python can run on different operating systems, including Windows, Mac, and Linux.

3.3.2 Javascript

Javascript is a widely used programming language for developing web applications. The advantages of Javascript are:

- Real-time interaction: Javascript allows for creating highly interactive and responsive user interfaces for web applications, which is what we needed for managing drone data in real-time.

- Multi-platform compatibility: Javascript can run on different operating systems and web browsers.
- Large community: Javascript has a large community of developers, which means that it is easy to find resources and tools to help with application development.

3.3.3 PyQt

PyQt is a graphical user interface library for Python. It allows developers to create rich and interactive graphical interfaces for their applications. We chose PyQt for its advantages, which are:

- Rapid development: PyQt is easy to learn and use, which can speed up the application development process.
- Rich user interface: PyQt allows for creating attractive and highly interactive graphical interfaces, which enhances the user experience.
- Multi-platform compatibility: PyQt can run on different operating systems, including Windows, Mac, and Linux.

3.3.4 Leaflet

Leaflet is an open-source mapping library for web applications. It allows developers to add interactive maps to their web applications. Its advantages are:

- Lightweight and fast: Leaflet is lightweight and fast, which means that it can run efficiently on the most common web browsers.
- Multi-platform compatibility: Leaflet can run on different operating systems and web browsers.
- Customizable: Leaflet offers many customization options for maps, including the ability to add custom layers and modify the appearance of maps.

3.4 Versioning

During the period of development of this project, we all had to work separately on different tasks but on the same project. In order to organize the merging of all the work and to increase our productivity and efficiency, we needed to work with a versioning tool.

Versioning is a critical aspect of software development that allows developers to track and manage changes to their code-base over time.

The purpose of versioning is to provide a structured system for developers to collaborate on and manage their code-base, ensuring that changes are tracked, documented, and easily reversible if necessary.

In order to work together more efficiently, avoiding conflicts and ensuring that everyone is working with the most up-to-date code, we used GitHub.

For this purpose, we created three main branches:

- Master branch which holds the official release history and the release tag that identifies the different releases.
- Dev branch contains all new features being developed and merged.
- Feat branch which spawns feature branches following a naming convention : feat/feature-Name.

4 Architecture and Results

In this section, we detail the specific implementation of each part of the project. This includes the architecture, design and final demonstration of each part of the project.

4.1 GUI

In our project, we use PyQt as the framework for the GUI. As previously described, PyQt is a GUI library for Python. It allows developers to create rich and interactive graphical interfaces for their applications.

PyQt's signal and slot mechanism is a key feature for handling user interactions with the graphical user interface. Signals are emitted when an event occurs, such as a button press or text input. Slots are functions that are connected to signals and are executed when a signal is emitted. This provides a convenient way to respond to user interactions with the GUI. One of the key benefits of the signal and slot mechanism is that it enables the creation of responsive and interactive GUIs. When a signal is emitted, the corresponding slot is executed immediately, which allows for real-time updates to the user interface.

We have analysed the features of the whole project by dividing the GUI into sections according to their functionality as follows:

- Main Windows & Main Widget
- Edit Area Widget
- Edit Line Widget
- First Simulation Widget
- Second Simulation Widget

We will discuss them in further detail in the following sub-chapters. Before that, allow me to introduce you to a visual UI design suite provided by PyQt, **Qt Designer**. Besides, all GUIs in our project are visualized finished with this suite.

Qt Designer is the Qt tool for designing and building graphical user interfaces (GUIs) with Qt Widgets. We can compose and customize our windows or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and test them using different styles and resolutions.

Widgets and forms created with Qt Designer integrate seamlessly with programmed code, using Qt's signals and slots mechanism, so that we can easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code. Furthermore, features like widget promotion and custom plugins allow you to use our own components with Qt Designer.

The file done through Qt Designer is a file with the extension `.ui`, and we can convert this structured file makeup into a Python file using the `pyuic5` command. For example:

```
pyuic5 -o test.py test.ui
```

4.1.1 HTML file containing maps

At the beginning of the project, when we analysed the requirements, we found that the maps we needed were basically implemented in JavaScript. This required us to embed a browser in the project to render the JavaScript libraries and code related to the maps.

Leaflet.JS is a popular open-source JavaScript library for interactive maps. It provides an easy-to-use API for developers to create and customize maps. By combining Leaflet.JS with Qt Channel, a powerful tool can be created to visualise geospatial data in PyQt applications.

Qt Channel is a mechanism provided by PyQt to enable communication between Python and JavaScript. It allows Python and JavaScript to exchange data and function calls seamlessly, providing an easy way to interact with web technologies in a Python application.

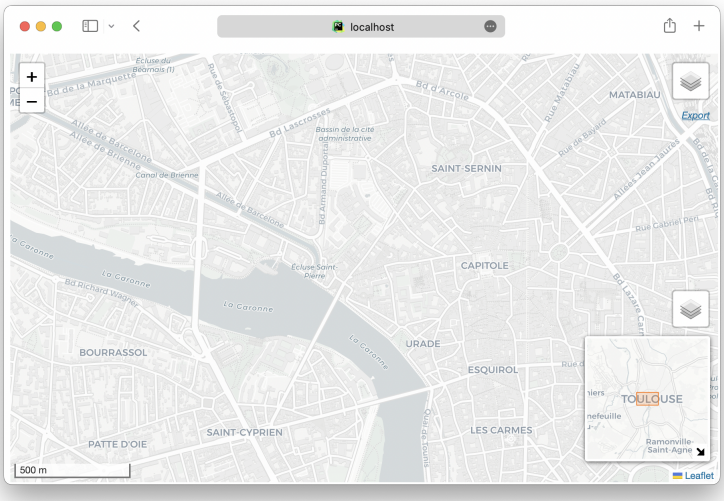

Figure 3: index.html

4.1.2 Main Windows & Main Widget

We have designed the main windows to look like the layout shown below:

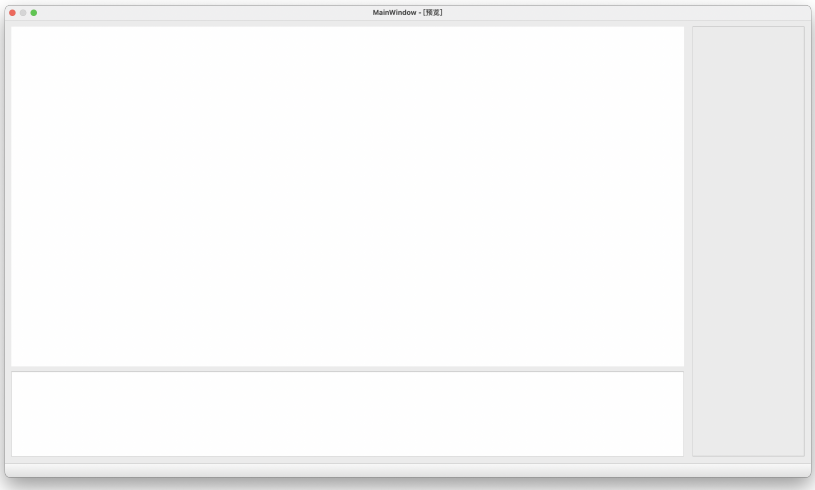

Figure 4: Main Window in designing

Obviously, the layout is generally divided into three parts:

- the largest area on the top left is a QtWebEngineView component,
- the bottom left is a textBrowser for displaying the system status,
- the right is a reserved Frame for binding widgets that implement different functions.

PyQt provides a class QtWebEngineView, which is a widget for displaying web content. It is based on the Chromium web engine and can be used to display web pages, HTML documents, PDF files and other content.

As mentioned before, to the right of the Main Window we have reserved a Frame to bind widgets that implement different functions. now we give the design of the first widget, Main Widget. It has no specific function, it just displays some information about the project name, the author, etc.

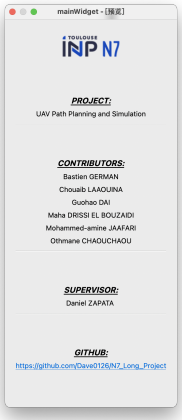

Figure 5: Main Widget in designing

The completed `index.html` is rendered into PyQt via WebEngineView. We get the final main window as shown below:

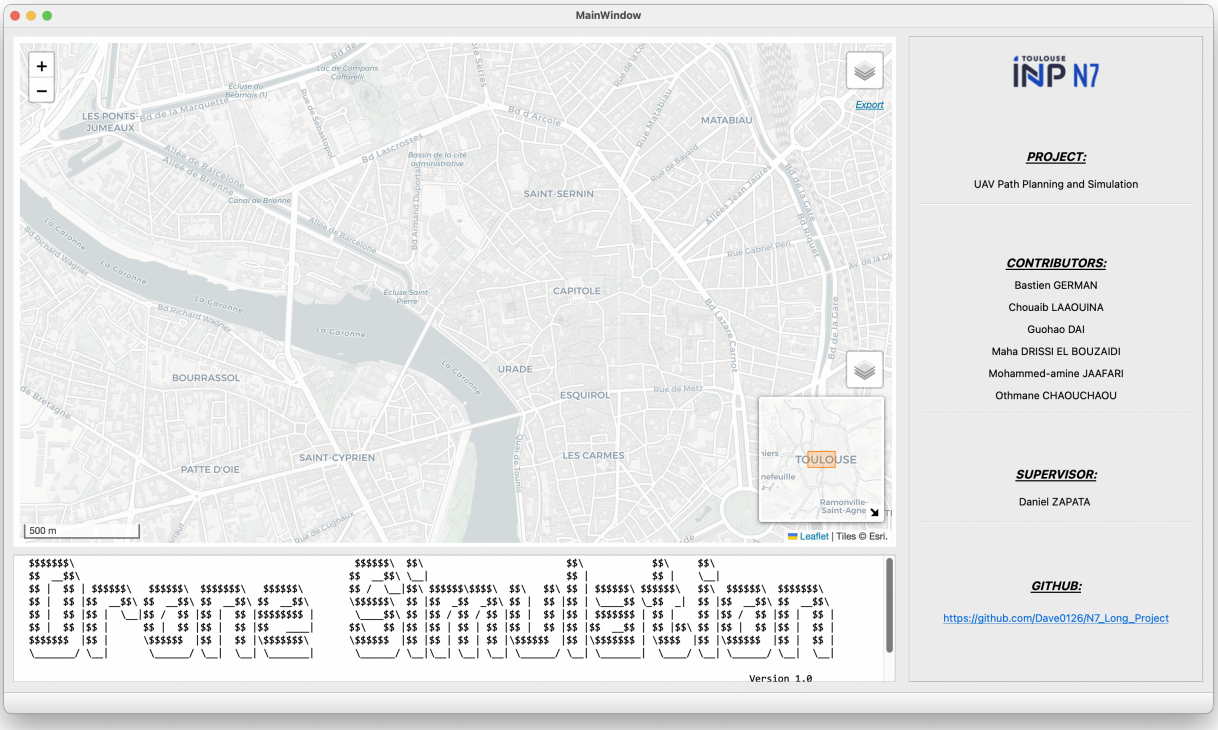

Figure 6: Main Widget in demonstration

4.1.3 Edit Area Widget

In this part of the design, the main feature of the Edit Area Widget is to add different no-fly areas to the map and add the following properties to each no-fly area.

- Name of area
- A brief description of the area
- The attributes of the area (in later advanced development the area should be divided into several different attributes to perform different no-fly operations. However, in the current version, the attribute is meaningless)

Also we would like to be able to display the GeoJSON information of the area added to the map and to export it as a `.json` file. We design the Edit Area Widget as requested above.

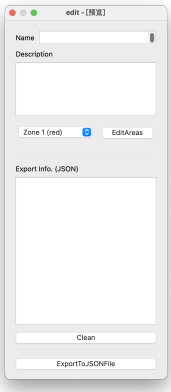

Figure 7: Edit Area Widget in designing

Similarly, the completed `index.html` is rendered into PyQt via WebEngineView. This gives us a functional demonstration of the Edit Area Widget, as shown below.

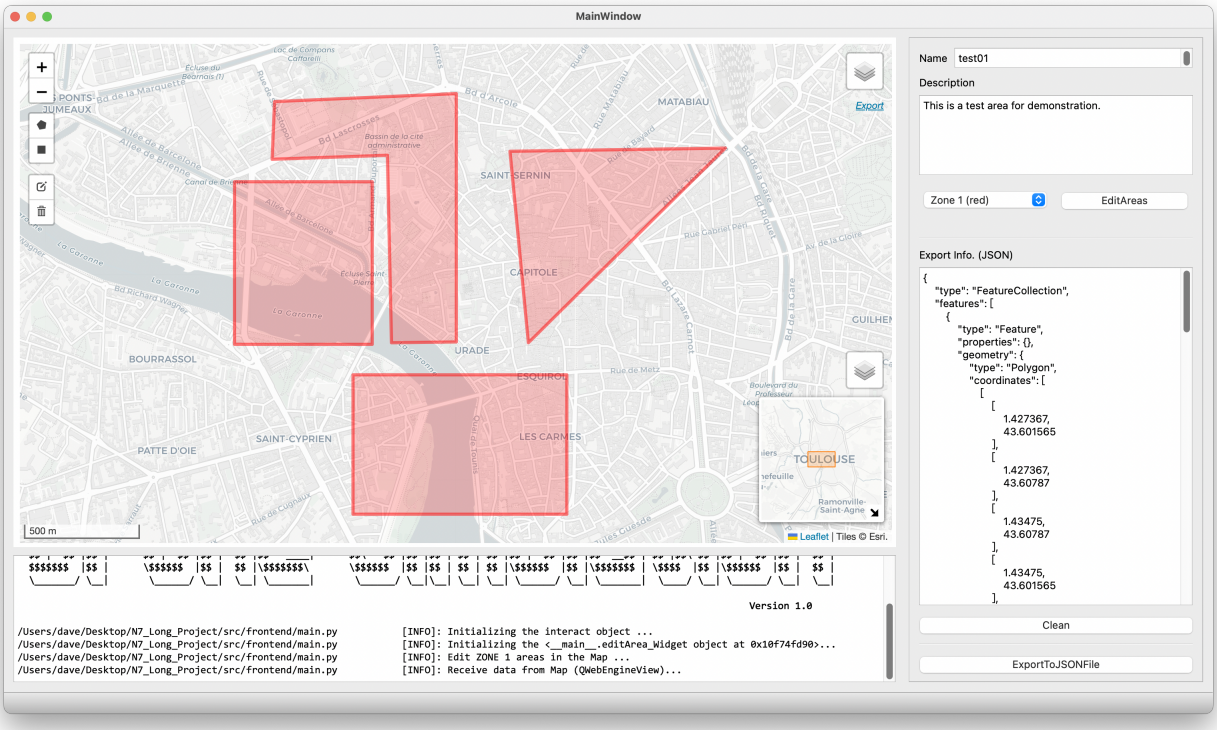

Figure 8: Edit Area Widget in demonstration

4.1.4 Edit Line Widget

The main function of the Edit Line Widget in this section of the design is to allow the user to manually plan a flight plan route by clicking within the map and adding the following attributes to each flight plan.

- Drone ID
- Planned flight time

We also wanted to be able to display the GeoJSON context for the routes added to the map and export it as a `.json` file. We designed the Edit Line widget as requested above.

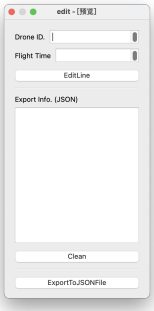

Figure 9: Edit Line Widget in designing

As before, the completed `index.html` is rendered into PyQt via WebEngineView. This gives us a functional demonstration of the Edit Line Widget, as shown below.

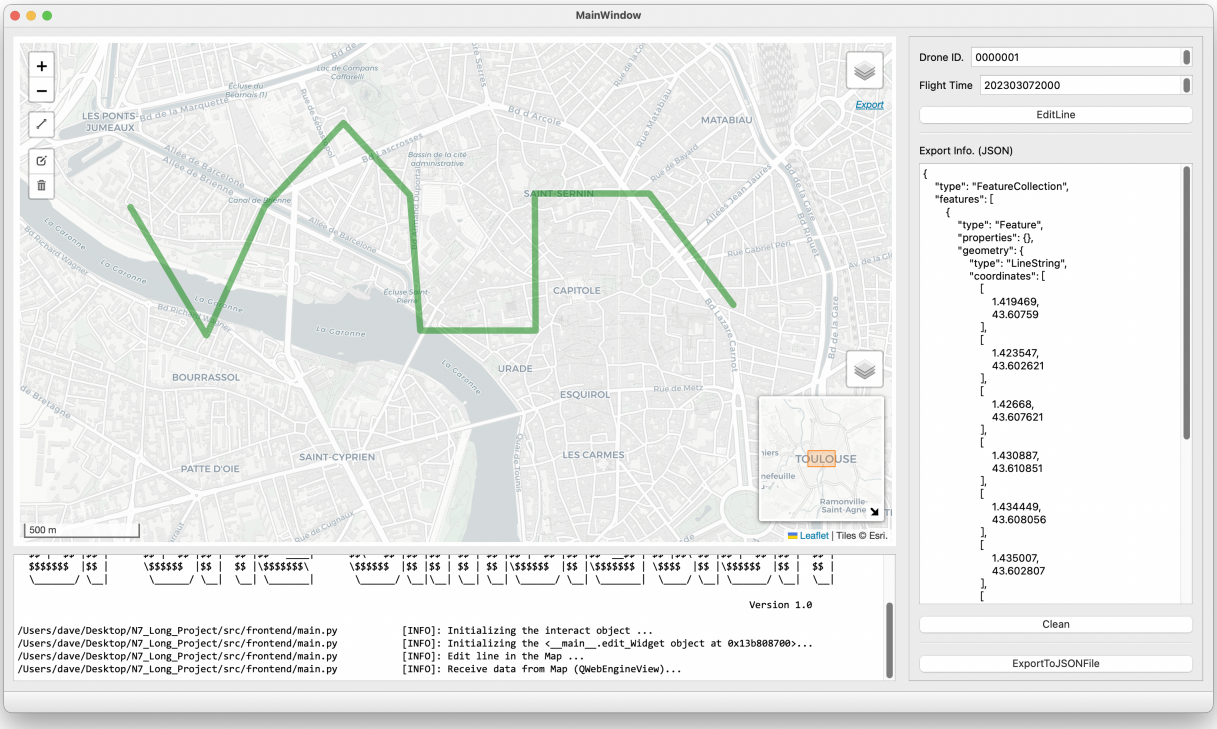

Figure 10: Edit Line Widget in demonstration

4.1.5 First Simulation Widget

In this part of the design, the main function of the First Simulation Widget is to run a Simulator thread in the thread pool to execute the flight plan route that the user has manually planned before. The Simulator thread sends the current position of the simulated drone to the receiving thread via scoket. In the First Simulation Widget there is a timed auto-sampler that is used to periodically mark the current position of the drone on the map.

We designed the First Simulation Widget to meet the above requirements.

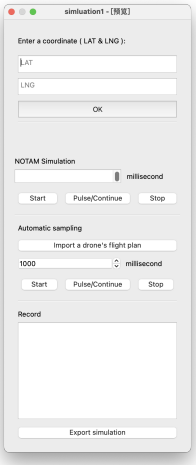

Figure 11: the First Simulation in designing

As before, the completed `index.html` is rendered into PyQt via WebEngineView. This gives us a functional demonstration of the First Simulation Widget, as shown below.

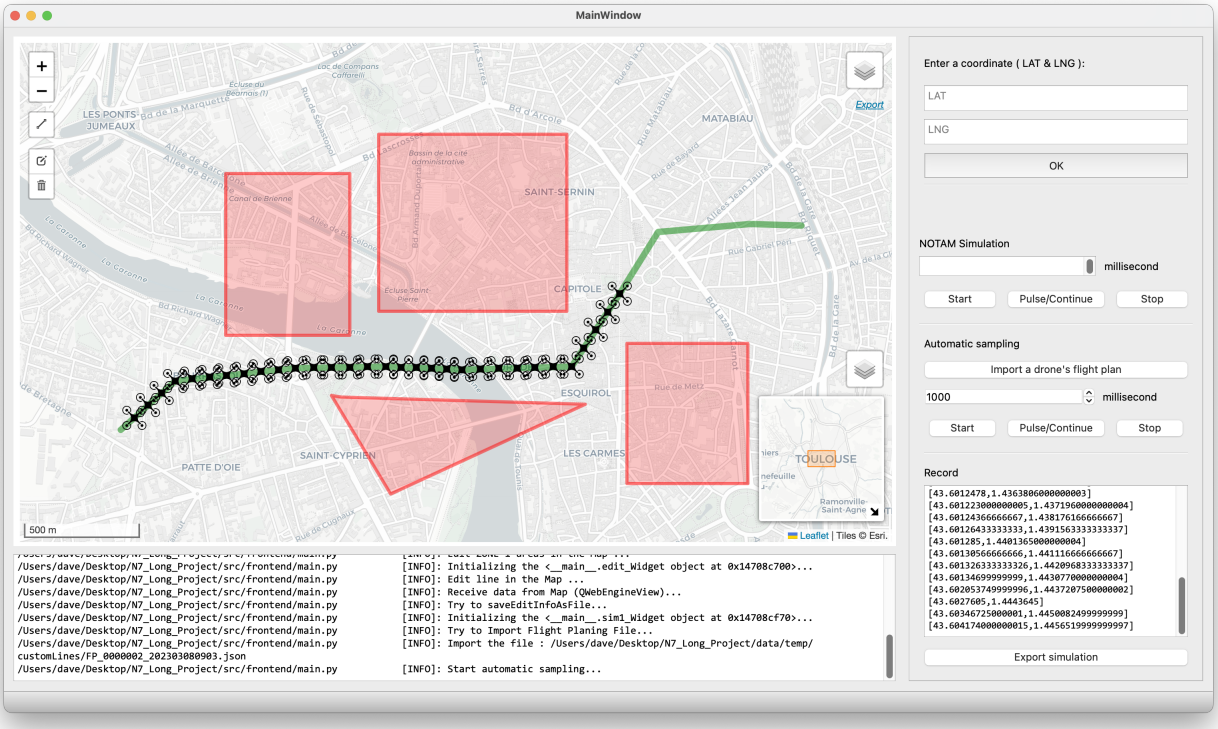

Figure 12: the First Simulation in demonstration

4.1.6 Second Simulation Widget

The main function of the Second Simulation Widget in this part of the design is that by selecting the start and end points on the map, our algorithm automatically calculates the shortest route that does not pass through a prohibited area and saves it locally as a flight plan route. A simulator thread is then run in the thread pool to execute the flight plan route. The simulator thread sends the current position of the simulated drone to the receiving thread via scoket. In the Second Simulation widget there is also a timed auto-sampler that is used to periodically mark the current position of the drone on the map.

We have designed the Second Simulation Widget to meet these requirements.

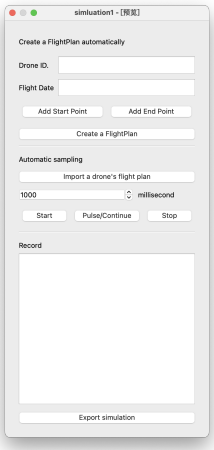

Figure 13: the Second Simulation in designing

As before, the completed `index.html` is rendered into PyQt via WebEngineView. This gives us a functional demonstration of the Second Simulation Widget, as shown below.

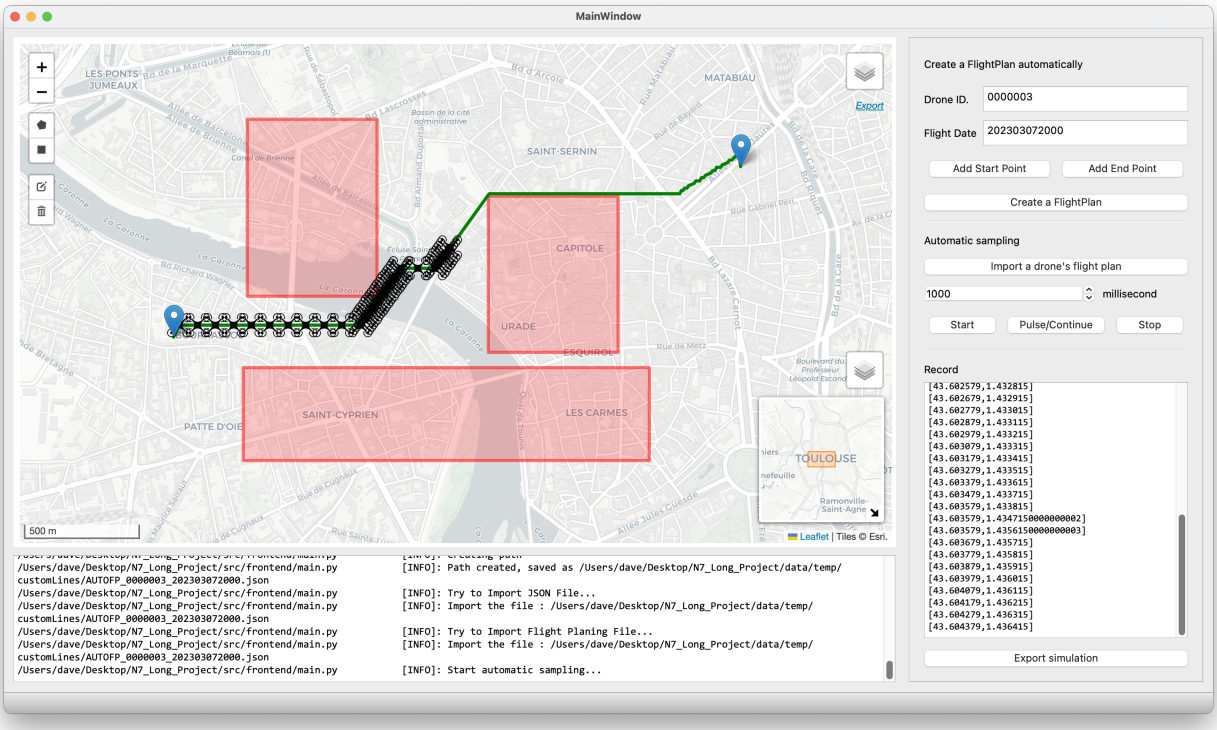

Figure 14: the Second Simulation in demonstration

4.2 Drone & Simulation

We have designed this section as a flight plan based drone flight simulation.

The `Drone` class represents a drone and contains methods to set its starting position and move it to the target position.

The `simulator` class reads the flight plan file in JSON format, extracts the coordinates from it, and moves the drone to each coordinate in turn. The `start_position()` method sets the starting position of the drone. `move_to()` calculates the distance between the drone's current position and the target position, and then calculates the number of steps required to reach the target position. The drone's position is then iteratively updated, the number of steps is added to the current position and the new position is displayed using a socket connection to the GUI.

The `RcvCmdThread` and `StatusReceiverTask` classes are used to receive commands from the front end and to update the status of the drone.

4.3 Area

In the current version, we use the `shapely` library to assist us with automatic path planning for the logic model.

`shapely` is a Python library for geometric operations using the GEOS library, which is the same library used by the widely-used GIS software PostGIS. It provides a high-level interface for working with geometric objects such as points, lines, and polygons. Some of the benefits of using `shapely` include:

- Easy to use: `shapely` has a simple and intuitive API, making it easy to use for a wide range of geometric operations.
- Fast and efficient: `shapely` uses GEOS, a C++ library for performing computational geometry operations. This makes it much faster and more efficient than Python-based implementations.
- Interoperability: `shapely` can be used with other Python libraries such as GeoPandas, PySAL, and Fiona to perform more complex spatial analysis.
- Comprehensive documentation: `shapely` has comprehensive and well-written documentation, making it easy for developers to learn and use the library.

However, considering the increased functionality of the logic at a later stage, we needed to handle each Area at a more fine-grained level. In the meantime, we have designed the classes for geometric shapes and `Zone` class.

Similarly to `shapely`, we have designed a series of methods for the geometry classes:

- `read_json()` takes the path to a JSON file and returns the features in that file.
- `onLine()` and `isIntersect()` methods can be used to determine if a point is on a line and if two lines intersect, respectively.
- `extract_zones()` extracts zones from a JSON file and adds them to the list of zones.
- `distance()` function that calculates the distance between two points.
- ...

4.4 Algorithm

4.4.1 A* algorithm

The A* algorithm, also known as the A-star algorithm, is a heuristic search algorithm that uses the actual cost from the source node to the current node and the estimated cost from the current node to the destination node to determine the next node to explore. The estimated cost is calculated using a heuristic function that estimates the remaining cost of reaching the target node. The heuristic function must be admissible (i.e., it never overestimates the actual cost of reaching the target node) and consistent (i.e., satisfies the triangular inequality)

Besides the heuristic, the A* algorithm uses a priority queue to keep track of the nodes to be explored. The priority of each node in the queue is determined by the sum of its actual and estimated costs. The A* algorithm maintains a closure set to keep track of nodes that have already been explored and ensures that each node is explored only once

4.4.2 Implementation of A* Algorithm in Grid

All components in Leaflet.JS (string line, polygon, etc.) are vector data consisting of points and the line segments between them. The authors have designed an algorithm that makes it possible to sample this modest data into the grid at a certain resolution.

In our program, the elements are converted into polygons of the shapely library by reading the GeoJSON file. If the polygons are of type "Polygon" or "MultiPolygon" they will be added to the `obstacles[]` list.

We can quickly get the global bounds with the `shapely.bounds()` method. With the bounds we can calculate the number of rows and columns needed for a given resolution of the grid and initialise the grid with `0`. It then calculates the index of each point on the grid and sets the value of the corresponding cell in the grid to

- `s` (if the point is the start point),
- `d` (if the point is the end point),
- `1` (if the point is inside an obstacle)

We consider the GeoJSON file below, as shown in Figure 14, suppose there is a square area with four points with WGS coordinate systems in JS are (1.435, 43.595), (1.445, 43.595), (1.445, 43.605) and (1.435, 43.605). Two points (1.43, 43.6) and (1.45, 43.6) on the left and right side respectively.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [ 1.43, 43.6 ],
        "type": "Point"
      }
    },
```

```
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [ 1.45, 43.6 ],
        "type": "Point"
      }
    },
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          [
            [ 1.435, 43.595 ],
            [ 1.445, 43.595 ],
            [ 1.445, 43.605 ],
            [ 1.435, 43.605 ],
            [ 1.435, 43.595 ]
          ]
        ],
        "type": "Polygon"
      }
    }
  ]
}
```

Figure 15: A square area and 2 points

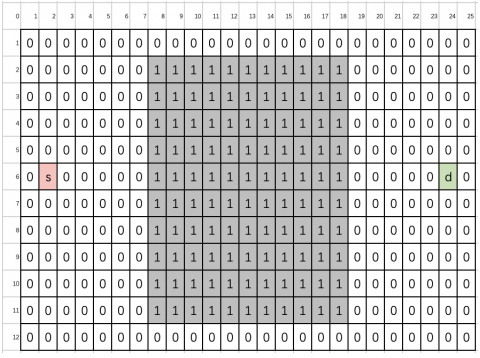

Figure 16: Sampling a square area and 2 points in Grid

Figure 15 shows the square area and the two points after grid sampling. where character “1” denotes the square region, “s” denotes the source and “d” denotes the destination.

Each node represents a location in the grid, and each edge represents a valid move between adjacent nodes. The actual cost of moving from one node to another is usually assumed to be one.

The most commonly used data structure for implementing the A* algorithm is the priority queue, which allows nodes to be explored in order of their estimated cost to reach the goal.

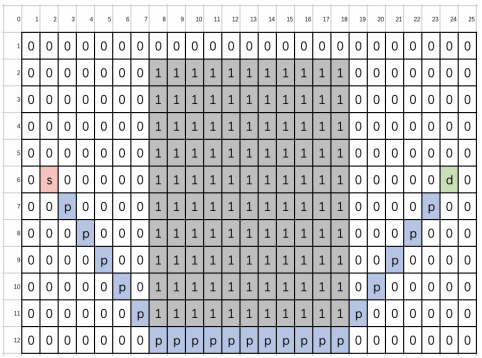

Figure 17: Path Planning by A* algorithm

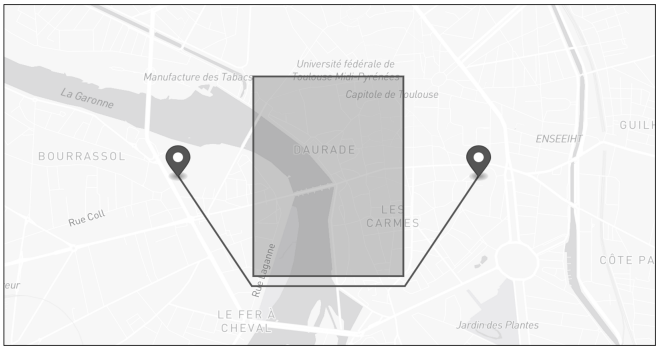

Figure 18: Show path on the map

5 Conclusion

The integration of drones into airspace safely and efficiently through the U-space framework is crucial to their widespread use in industries such as aerial photography, agriculture, and disaster management. Indeed, many issues could arise from the use of drones such as : having drones colliding with objects or other drones, or using drones in unauthorized areas for surveillance, which raises privacy concerns.

Therefore, the development of the U-space framework is an important step towards the safe and efficient integration of drones into airspace. The technical project presented here, which involves the design and development of a demonstrator for modifying a drone's trajectory in response to airspace reconfiguration, is a practical example of how U-space can be implemented to ensure the safety of drone operations in emergency situations. Thus, Our project represents the first step towards efficiently dealing with this aspect of integrating drones as common agents in the airspace.

As the use of drones continues to expand, the need for innovative solutions to ensure their safe and effective integration into airspace will become increasingly important. The integration of these innovative solutions is an aspect that could be improved in our project. Possible improvements could be for example :

- The processing and simulation of multiple drones at the same time.
- The use of different algorithms from A* according to the situations where they could be more optimal.
- The use of a notification system if there is a need for the intervention of a person.
- The transition to 3D to be able to visually represent the differences between the altitudes of the drones.

Overall, the prospects for the implementation of the system in the context of intelligent mobility in cities are also promising. The use of drones for tasks such as delivery and surveillance could help to reduce traffic congestion and improve public safety. The continued development of drone technology and the implementation of innovative solutions such as U-space and the drone path configuration system will be crucial to unlocking the full potential of drones and ensuring their safe and effective use.

6 Bibliography

- "U-space." SESAR Joint Undertaking, https://www.sesarju.eu/U-space. Accessed 8 March 2023.
- "Commission Implementing Regulation (EU) 2021/664." EASA, 23 April 2021, https://www.easa.europa.eu/en/document-library/regulations/commission-implementing-regulation-eu-2021664. Accessed 8 March 2023.
- "U-Space study." European Defence Agency, 8 January 2021, https://eda.europa.eu/U-Space-study. Accessed 8 March 2023.
- "U Space Concept - iConspicuity for GA & Rotorcraft in U-space and beyond." EASA, https://www.youtube.com/watch?v=ITXb3uBwbZk&list=PLTfS24aKkJn76qYVtUfhswviiJSJHCl8Z&index=3. Accessed 8 March 2023.
- "SESAR U-space webinar : Concept of Operations." SESAR, https://www.youtube.com/watch?v=68zv-neUOFE&list=PLJItpHUetWvHdjp2MarchYUWqdwJ0r8ZAXudps&t=678s. Accessed 8 March 2023.
- "SESAR JU PODIUM Project - Demonstration by NLR in Eelde (2019)." SESAR, https://www.youtube.com/watch?v=2msVXq6a1Zg. Accessed 8 March 2023.
- "U-Space services implementation monitoring report." EuroControl, https://www.eurocontrol.int/publication/u-space-services-implementation-monitoring-report. Accessed 8 March 2023.

7 Access to the Github Repository

Link : https://github.com/Dave0126/N7_Long_Project