Abstract of project: UAVs Path Planning and Simulation

Guohao DAI

The National Institute of Electrical engineering, Electronics, Computer science, Fluid mechanics & Telecommunications and Networks, 31071 Toulouse Cedex 7, FRANCE

9 March, 2023

Abstract

This article presents the author, Guohao DAI, on his long project from February to March 2023. The work was carried out remotely online, cooperating with colleagues who are both specialising in Logiciel, to develop a Graphical User Interface (GUI) with path planning and simulation of UAVs in diverse airspace. The author worked as a code developer in the team, completing all GUI parts, including the PyQt framework and map-related JavaScript. Besides, mapping the vectorial geographic coordinate system to the Grid's X-Y integer coordinate system with a certain resolution missing through the study of the Grid-based A* algorithm. In addition, the author passionately collaborated with other colleagues to fix issues with the integration of individual modules between projects.

In two months of full-time work, the team delivered a release. Furthermore, issues found during testing were fixed to enhance the stability of the system. By analysing the requirements, not only updated for future versions planned but feasibility verification and trial development were also carried out.

Keywords: Drone, Path planning, A* algorithm, PyQt

1. Introduction

As shown in Figure 1, the GUI of the project is divided into two main parts in general: the part on the left side is the map-related area and written in JavaScript. The second part on the right side is the control widget written in Python. There are four groups in total, which can be switched according to different functions. Thus, the communication of data between two different languages becomes a challenge.

This article will focus on the bi-directional channel between PyQt and Leaflet.JS as well as the implementation of the A* algorithm.

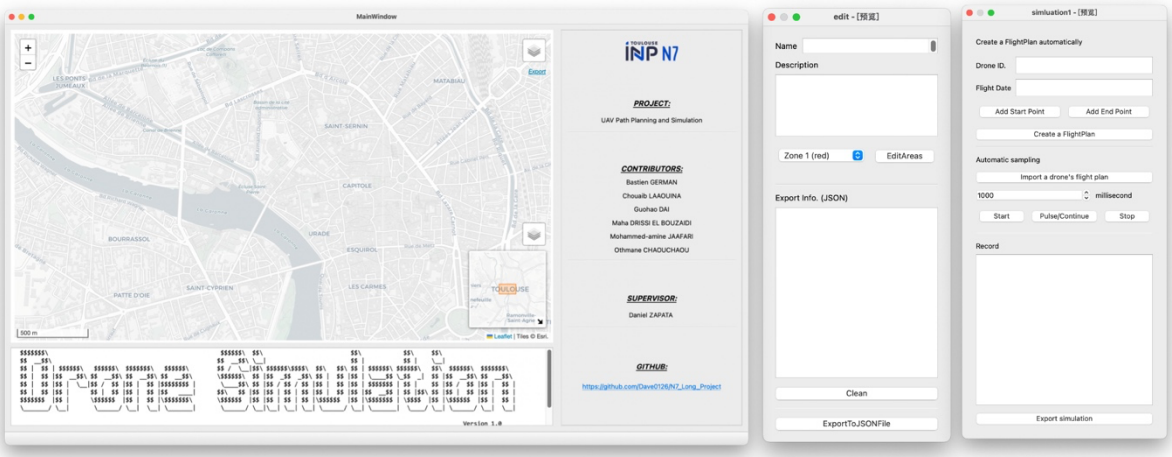

Figure 1: An overview of the different widgets in the GUI

2. Methodology and Practice

2.1 Signal and Slot Mechanism in PyQt [1]

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.

2.2 QtChannel and Leaflet.JS [2]

QtChannel 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.

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 QtChannel, a powerful tool can be created to visualise geospatial data in PyQt applications.

2.3 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[3] 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.

2.4 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.

As shown in Figure 2.1, 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.

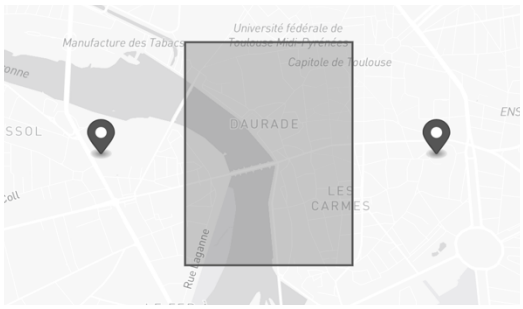

Figure 2.1: A square area and 2 points

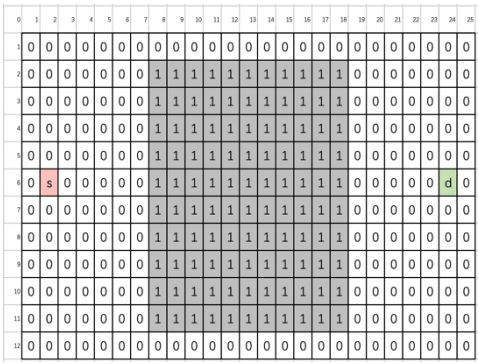

Figure 2.2: Sampling in Grid

Figure 2.2 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.

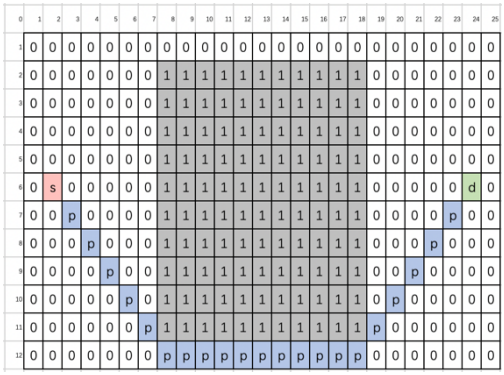

Figure 2.3: Path Planning by *A** algorithm

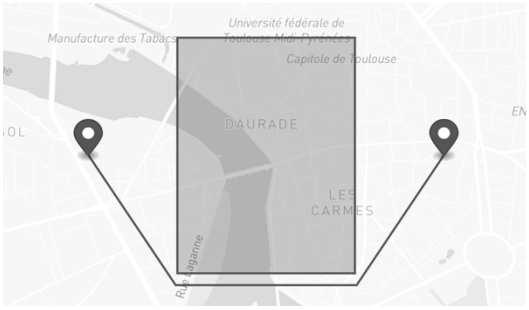

Figure 2.4: Show path on the map

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.[4]

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.

3. Conclusions

Through the completion of the project, the following conclusions were reached:

- The integration of PyQt with WebEngine and Leaflet.JS provides a powerful way to create interactive desktop applications that incorporate web-based functionality.
- By using the A* algorithm in a grid, we can find a path between two points on a map while avoiding obstacles represented by polygons. This approach has applications in a variety of fields, including robotics, video games, and geographic information systems.
- The project has not only trained the author's ability to collaborate, communicate with others, but also to learn numerous new domains and knowledge through group discussions.

4. Acknowledgement

The team would like to express their sincere gratitude to each member for their cooperation and dedication. Their collective efforts and contributions have led to the success of the project. Special thanks to the supervisors for their invaluable guidance and support throughout the process.

References

[1] PyQt, retrieved 10 February, 2023 from https://www.riverbankcomputing.com/static/Docs/PyQt5/
[2] Leaflet.JS, retrieved February 10, 2023 from https://leafletjs.com/
[3] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100-107.
[4] Choset, H. (2005). Principles of robot motion: theory, algorithms, and implementations. MIT press.