Sabancı Universitesi

FACULTY OF ENGINEERING AND NATURAL SCIENCES

Mechatronics Engineering and Computer Science & Engineering

PROJ 302 Summer Internship - GUI Design for Automated Mission Planning and Controler Synthesis

Edin Guso
University of Texas at Austin

1 July - 29 August 2019

Project Summary

Mission planning and controller synthesis for autonomous robots is a complex task and requires knowledge about several topics such as control, robotics and linear temporal logic. Wrapping these tasks with a graphical user interface and making these tools usable by people without the technical knowledge is important. Because that would result in an increasing number of people getting involved and thus make the field grow.

The main objective of my internship in Autonomous Systems Group at the University of Texas at Austin was to create this graphical user interface. The constraints for this software were that the software had to be easy to use, include all the necessary features, handle all the mission data in an efficient way and also be able to output this data in a meaningful way so that the controller synthesis software could take it as the input.

In addition to developing the mission planning software, a path planning software has been developed to process the data from a simple mission to confirm that everything works as intended. Finally, a simulation of those paths has been created to have a demonstration of the whole process, from the user inputs to the simulation.

Company Information

University of Texas at Austin is a well-known university in the United States with more than 51000 students and 3000 teaching faculty [1]. The university was founded in 1883 [2]. It is one of the highest ranked universities. The core purpose of the university is "to transform lives for the benefit of society." [3].

I conducted my research in the Oden Institute for Computational Engineering and Sciences. The institution was established in 2003 and has more than 330 people [4].

I was a part of Autonomous Systems Group. The group consists of 18 PhD students and several post-docs and is lead by Prof. Ufuk Topcu. The group mainly does work on machine learning, autonomous robots and formal methods.

Object Design

Since all the data about the missions had to be handled efficiently, there was a need for a well-planned class hierarchy. The most important issue with the object design was being able to hold all the different vehicles in a single container. Also, each vehicle had to hold all its missions (more importantly the shapes of the missions) in a single container. To do that the following class hierarchy was developed.

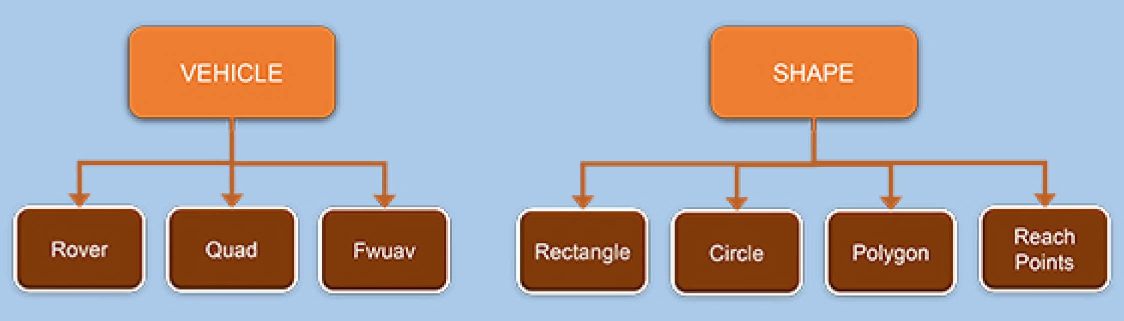

Figure 1: Class Hierarchy

It was a necessity to have these different classes and they had to be stored in a single container, therefore two base classes were designed, Vehicle and Shape. Then the necessary sub-classes were implemented by inheriting from these two classes. This made it possible to create different types of objects (eg. Rectangle or Circle) and store them in a single container (eg. vector<Shape>).

Click Detection Algorithm

Result: Returns the clicked vehicle or mission
Inputs: cursorPos; vehicles;

```
for each vehicle in vehicles do
    if vehicle.isInside(cursorPos) then
        return vehicle;
    else
        for each mission in vehicle.getMissions() do
            if mission.isInside(cursorPos) then return mission ;
        end
    end
end
return NULL;
```

Algorithm 1: Vehicle and Mission Detection on Mouse Click

Methods and Tools Used

The main software used throughout the project was Qt Creator. Qt is an open source widget toolkit and I have used it to create all the graphical user interface. It is a well-developed tool for developing graphical user interfaces. There are many existing libraries for widgets that can be used. Also, it is possible to inherit from the existing classes and implementing custom widgets. The code was written in C++. The version of Qt used was 5.13.0 with the compiler MinGW 7.3.0 32-bit for C++.

For the simulation part of the project, several other software has been used. ROS (Robot Operating System) and Px4 flight control software has been used for handling all the aspects of the robots such as the low-level control of the robots. Gazebo simulator, an open source 3D robotics simulator, has been used to simulate the physics and the environment in which the robots will move.

Path Planning

Result: Creates and fills the gridworld with the given vehicles and missions
Inputs: vehicles; missions, gridworld;

```
for each cell in gridworld do
    gridworld[row][col] ← 1;
end
for each vehicle in vehicles do
    for each cell in gridworld do
        if cell.isInside(vehicle.pos()) then gridworld[row][col] ← -1;
    end
end
for each mission in missions do
    if mission is reachmission then
        for each cell in gridworld do
            if cell.isInside(mission.pos()) then gridworld[row][col] ← 2;
        end
    else
        for each cell in gridworld do
            for each pixel in cell do
                if mission.isInside(pixel) then gridworld[row][col] ← 0;
            end
        end
    end
end
```

Algorithm 2: Vehicle and Mission Translation to Gridworld

Result: Computes all the necessary paths
Inputs: gridworld, source, reachpoints;

```
for each destination in reachpoints do
    aStarSearch(gridworld, source, destination);
end
for each start in reachpoints do
    for each end in reachpoints do
        aStarSearch(gridworld, start, end);
    end
end
```

Algorithm 3: Path Computation

Time Complexity:

$$O(MP + CN^2)$$

Results

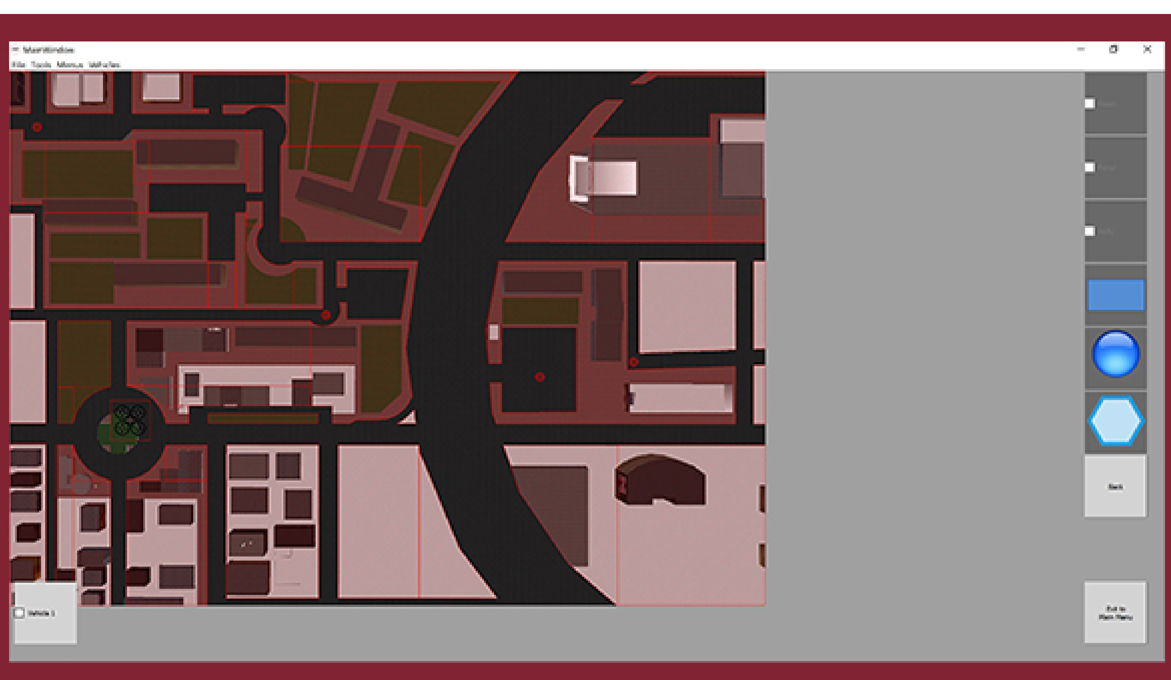

Figure 2: Example Mission

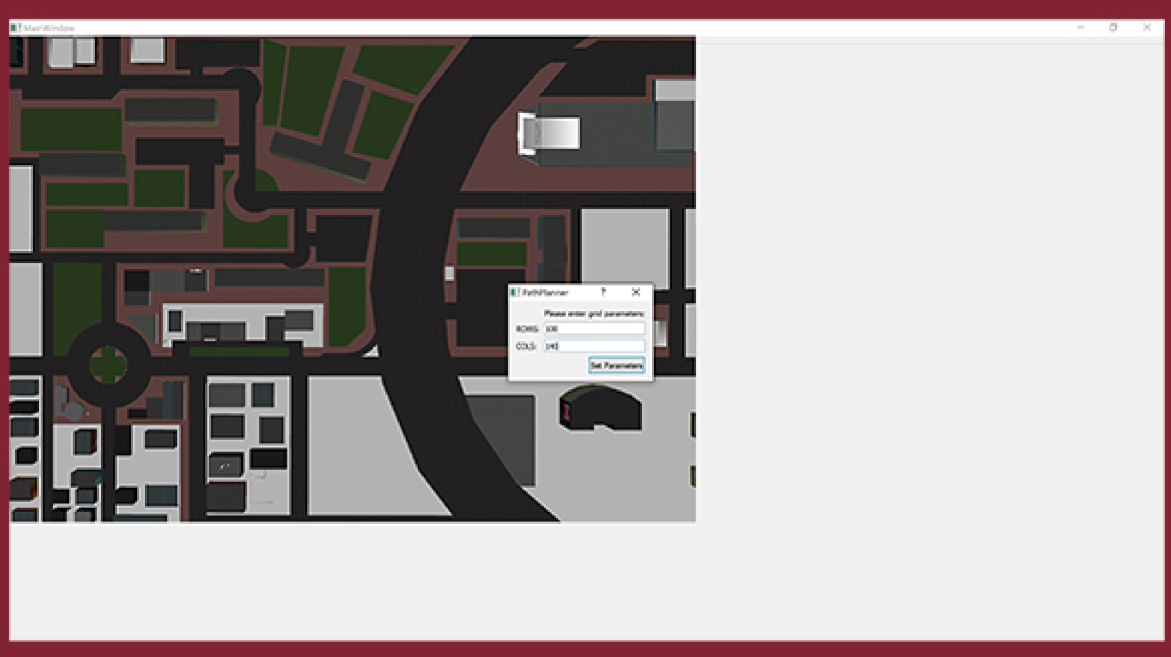

Figure 3: Gridworld Resolution Prompt

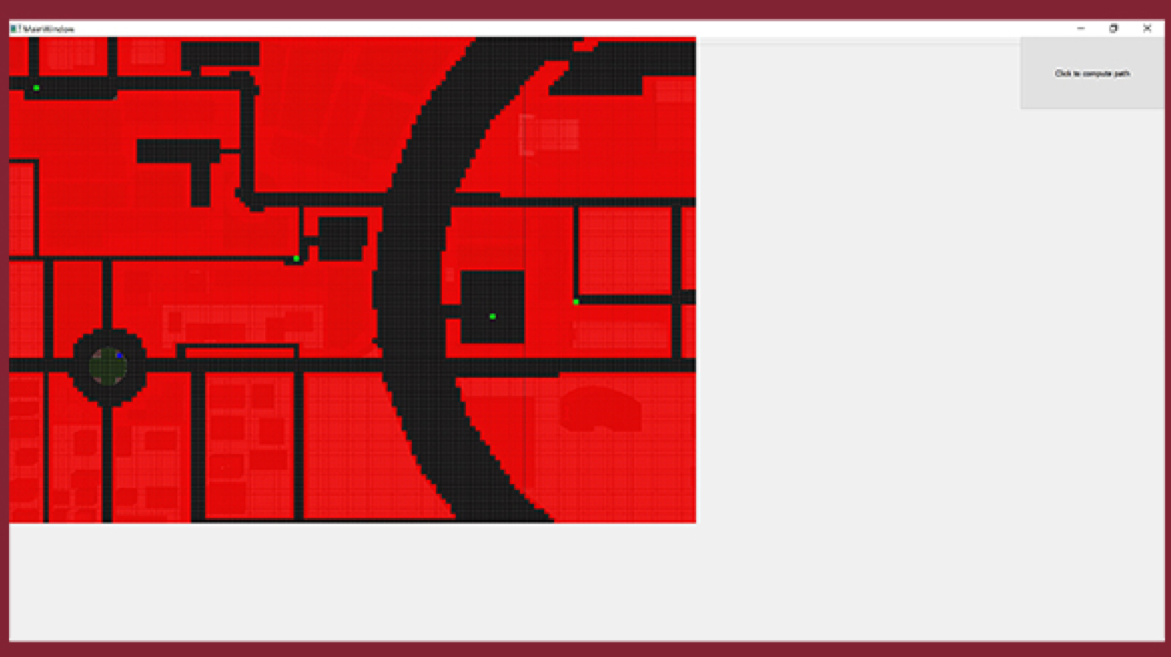

Figure 4: Gridworld Created

Figure 5: Example Path

References

[1] University of Texas at Austin. Overview. https://www.utexas.edu/about/overview. [Retrieved on 28 Aug 2019].

[2] University of Texas at Austin. Facts & figures. https://www.utexas.edu/about/facts-and-figures. [Retrieved on 28 Aug 2019].

[3] University of Texas at Austin. Mission & values. https://www.utexas.edu/about/mission-and-values. [Retrieved on 28 Aug 2019].

[4] Oden Institute for Computational Engineering and Sciences.Welcome to the oden institute. https://www.oden.utexas.edu/about/welcome/. [Retrieved on 28 Aug 2019].