Sentinel Drone Rulebook

blogpost-style , rulebook

codingcoffee 1 November 25, 2022, 7:27am

Sentinel Drone

1. Introduction

1.1 Background

From remotely heating a coffee pot to having an autonomous car, everything has resulted from the galvanization of complex engineering breakthroughs, which, while alleviating our mundane routine, has also caused a new rise in Grand Theft Auto-related cyber crimes.

This year, in eYRC 2022, we consider a similar scenario to tackle. With the technological improvements in the drone industries and amalgamation of drones with various GIS (Geo-spacial Information systems) techniques, we can help prevent grand theft auto using Sentinel Drones.

The drones will be used to canvas an area of land while simultaneously scanning it for the “fingerprint” (to simplify, we’ll be using a rectangular yellow box) of a stolen car. Now, mind that the drone knows its coordinates globally, using GPS in the real world and a ceiling-mounted camera in our simulated world. However, it needs to relay the location of the detected car and not the drone itself to the police. This is where GIS comes into play. The drone will use pattern matching on the land to detect the car’s location, estimate the coordinates and send the data back to the main server.

The toil you saw earlier is the case not only for catching the bad guys but any form of reconnaissance which needs to be conducted, whether in hyper local cases like warehouses for item detection 135 or more outdoor use like military applications in bomb detection, or for simply canvassing a particular area and numerous other places. Let us develop robots centred on reducing human labour and improving the quality of life of humans. We have designed this theme and our interactions such that at the end of this if you put in the time and effort, you’ll be better suited to develop these solutions yourself!

In the next section, we shall look at the detailed description of the final task of this theme!

1.2 Problem Statement

Our problem statement consists of an autonomous drone tasked with surveying and identifying events (denoted by colored boxes) placed on a map, using methodologies like image processing and uploading data to GIS (Geographic Information Systems) systems. You’ll first work on these tasks in simulation to

hone your skills, and finally learn to assemble a nano drone by yourself and execute the same tasks on hardware!

1.3 Tools Used

For solving our problem statement we shall use Betaflight ecosystem for controlling the UAV, Python for writing the autonomous stack, and ROS for integrating various aspects of autonomy required in the solution.

2. Theme Description

This theme is an abstraction of a surveying drone. It is responsible to survey a given area of land and search particular events (denoted by colored blocks) on the arena. After identifying/detecting the blocks, it needs to calculate the geo-coordinates of the detected block using GIS techniques and report the same. This entire task, from detection to reporting the geo-coordinates, has to be done as quickly as possible.

2.1. Terms

2.1.1. Sentinel Drone

The Sentinel Drone (SD) is the Unmanned Aerial Vehicle (UAV), in the formfactor of a quadcopter which is the surveying agent in the theme. It has a WhyCon marker mounted on top of it for easily locating the drone and to serve as a feedback mechanism for the control system to control and move the drone.

2.1.2. Yellow Block

These are placed randomly on the arena and denote events on the map. This event could be mapped to something like a fire in a building or a road accident. Each block will have to be detected by the drone and reported on a given ROS topic. Each box is a cuboid of the dimensions: **L = 5cm W = 5cm H = 5cm** A sample yellow box is shown in the below figure

Figure 1: Yellow Block

2.2 Inputs

The coordinates will be given to the participants just before the theme run. This is to ensure the coordinates cannot be hardcoded into the script. The coordinates are of the form [*row*][*column*] printed on the arena, e.g. B4, C2. The blocks must be placed in the given location to prepare the arena. For placing the box on a particular coordinate, e.g. C4, you can use strings to mark the intersection of **C** row and **4** column. And then remove the string once the block is placed.
Tip: One can use temporary adhesive to stick the block on the arena.

Refer to the below figure to place the block at a particular coordinate

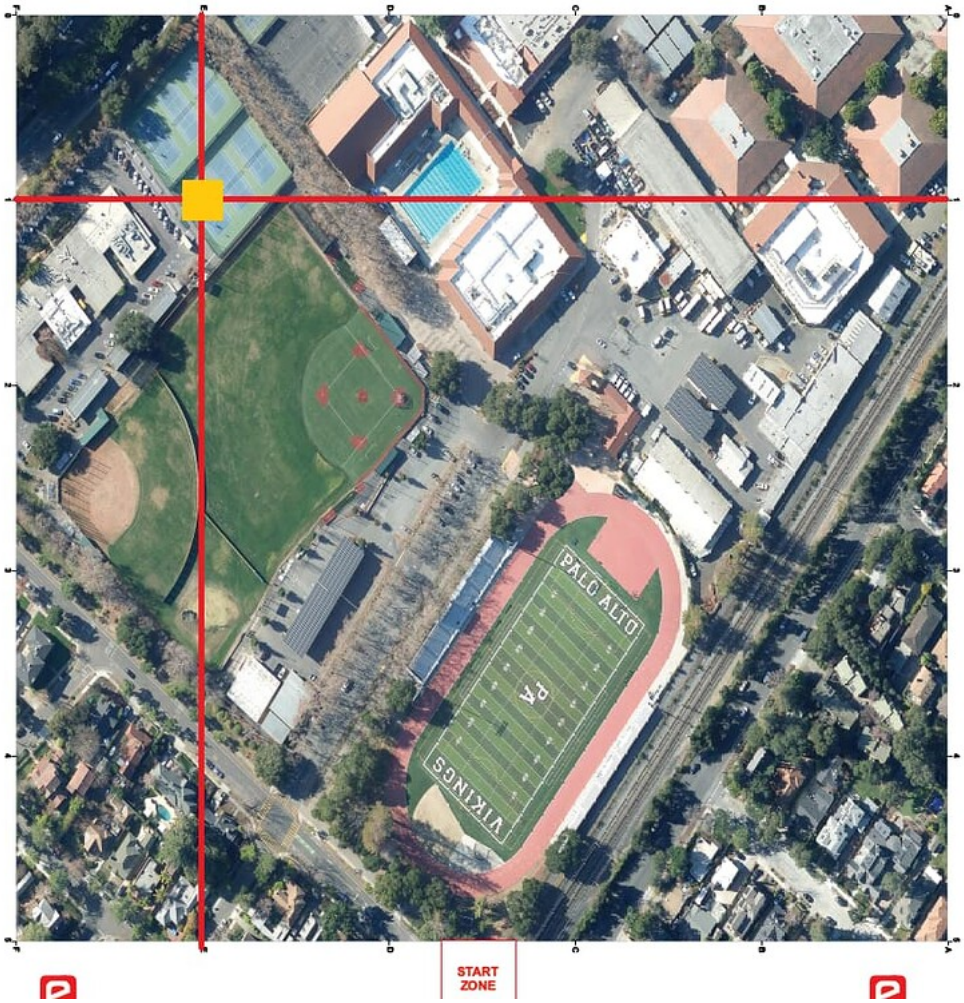

Figure 2 : Sample placing of block

3. Arena

Each team has to prepare the arena. Preparing the arena consists of:

1. Printing the Arena on the Flex Sheet
2. Preparing yellow blocks
3. WhyCon Marker Construction
4. Setting up the Overhead Camera

NOTE: Teams are not allowed to make any changes in the arena design. Any team making unauthorized modifications will be disqualified from the competition.

3.1 Printing the Arena on the Flex Sheet>WARNING: Please be careful while handling the flex sheet – avoid folding it like a bed sheet since the resultant folds will cause problems while the robot moves. One way of “flattening” flex if it has been compromised is to hang it for a few hours in the sun – it tends to straighten out. Never attempt ironing it or applying heat of any kind – it may be a fire hazard.

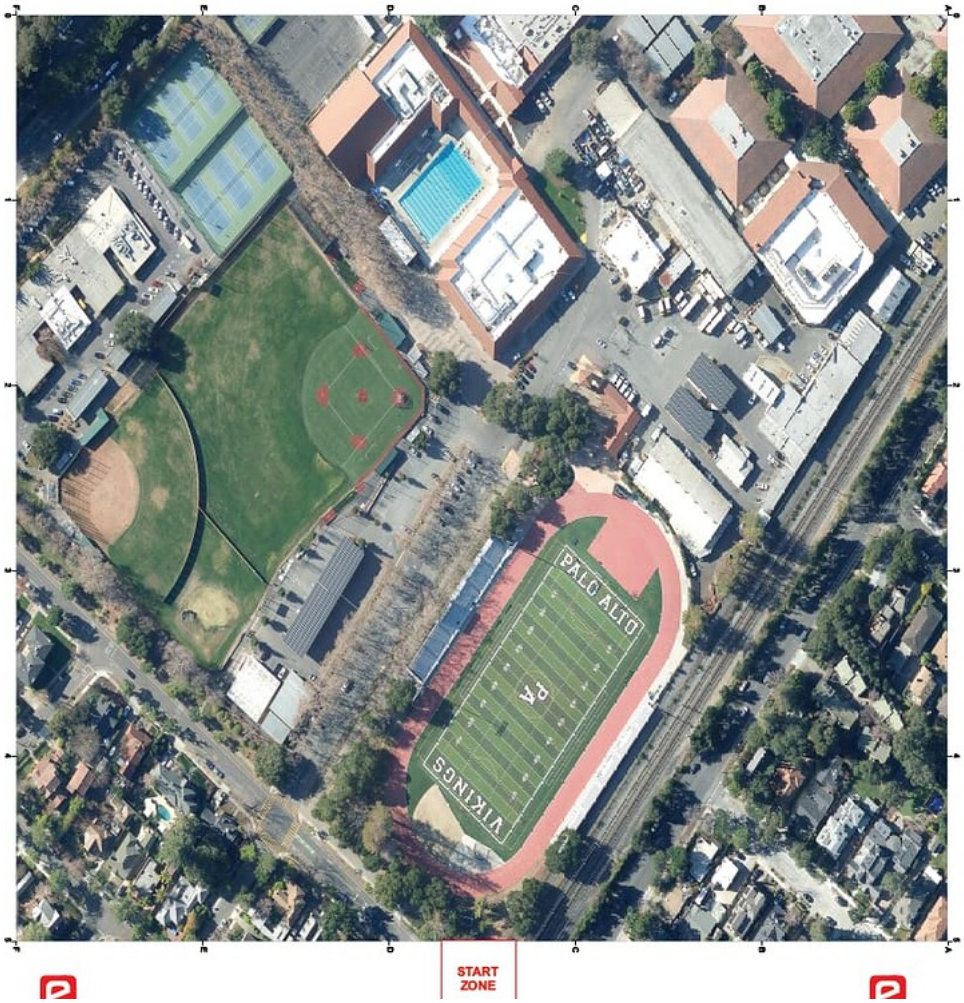

Figure 3: Arena

Dimension of the flex sheet is 7ft × 7ft.

3.2 Preparing the yellow block

A yellow block is an area where the drone is supposed to detect an event. Each block will be colored in a particular shade of yellow for the drone to identify and locate them easily from a height. Use **this** document to print yellow sheet of paper and wrap the box with yellow sheet to make the yellow box

3.3 WhyCon Marker Construction

You will find a printed A4 sheet of WhyCon markers in the package which has been shipped to you. Alternatively, you can download it **here**. You have to attach the WhyCon marker to the top of your Drone, so that the overhead camera can track it.

3.4 Setting up the Overhead Camera

You have been provided with a camera, and it’s USB extension cable. The provided camera must be mounted to have a complete view of the arena. The camera should be mounted above the center of the arena at a height of 8-10 feet. Make sure that you fix the camera parallel to the ground and not in an inclined manner. Teams are expected to use their creativity to design an arrangement to mount the camera, for example, hang it from the ceiling, construct a frame etc. The extension cable will connect the camera to the PC/Laptop.

4. Hardware & Software Specifications

4.1 Hardware Specifications

Hardware Specifications
All the participating teams must use only the components sent to them in the kit. The drone should be completely autonomous. The team cannot use any
wireless remote for its manual control. Teams are allowed to create any type of mechanical mount for mounting camera above the arena.

> Note: No other expansion and/or micro-controller based boards shall be attached to the drone

4.2. Ubuntu 20.04

- The operating system on which all the softwares run is Ubuntu 20.04.

4.3. ROS Noetic

- Robot Operating System (ROS) is the framework used to integrate the components in the theme. The version of ROS supported in the competition is Noetic

4.4. Python

- Python is the language in which the participant’s programs are written
- The version of Python supported with ROS Noetic is Python 3
- Hence, all the Python programs interfacing with ROS Noetic framework should be written in Python 3. Python 3 comes pre-installed with ROS Noetic

4.5. Additional packages and software.

- One can download and install additional software or packages required to perform a specific task after seeking approval from e-Yantra.

5. Theme Flow

- The arena is set up as per the given inputs.
- Participants will be given a ros-launch file that will start logging the necessary topics in the rosbag file.
- All the scripts and programs must be started before the run starts. No program/script can be started after the run starts
- The drone is armed at the start zone, indicating the start of the run. Time will be recorded from this point
- The drone surveys the entire arena finding the yellow blocks. The drone can send detection messages as and when it detects the blocks while surveying or at any other time later. The geolocation can also be sent as and when estimated or later at some other time before the end
- The drone has to detect and report the geolocation of as many blocks as possible in 250 seconds timeframe
- The drone lands on the start zone to indicate the end of the run

6. Theme Rules

- The run will be of the finite time of **250 seconds**.

- The drone has to complete as many block detections and report the geolocation coordinate of the block and land back in the designated area within this timeframe.

6.1. Takeoff/landing rules

- The drone should be kept within the start zone drawn on the arena. The drone should land in the same area; the WhyCon coordinates of the start zone will be captured at the start of the run, indicating the WhyCon coordinates of the takeoff and the landing zone.

6.2 Detection Rules

- When the drone detects a block, it needs to publish the event on the designated ROS topic
- The detection is considered valid only if the block is visible in the camera frame of the drone.

6.3 Geolocation Rules

- The Geolocation Coordinates are calculated for a block, it needs to be published on the designated ROS topic. The geolocations can be sent at any time frame, it can be sent immediately after detection or can be sent any time before/after the drone lands.
- The Geolocation Coordinate is considered valid if it is within a 3% margin of error

6.4 Re-positioning Rules

- Any kind of re-positioning is **NOT** allowed during the run. Once the drone is armed, no manual intervention is allowed.
- If the drone crashes on the ground or into any structure and is no more controllable by the algorithm, then it will indicate the end of the run. However, if after any kind of crash, the drone can still continue to operate without any manual intervention, then the run can continue.

7. Judging and Scoring Rules

The team’s total score is calculated by the following formula:

Score = (250 - t) + TB + (BD * 50) + (GL * 150) + Bonus

t : Time taken from taking off to landing/detecting and sending the geolocation of all the blocks (whichever is later)
TB : Takeoff bonus, the value of TB is 50 points
BD : Number of blocks successfully detected
GL : Number of Geolocations correctly identified for detected blocks
Bouns: Bonus component
Valid Run: A run is considered valid only when the drone takes off and **detects** at least 1 block.

Notes:

- At least one correct detection is required for a valid theme run. The geolocation coordinates need not be sent for a valid theme run.
- The bonus mentioned in the scoring formula above is worth 50 points and is granted only when the drone is able to **take off, detect all the blocks correctly, report the geolocation of all the blocks correctly, and also land in the designated area**.
- Every team will be given 2 attempts at flying the drone.

- If the drone malfunctions / crashes / goes out of the camera frame and is not able to come back into it, it is still considered as an attempt.
- The maximum time for a run is **250 seconds**
- The start time is considered from when the drone is armed.
- The end time is considered till the drone lands **or** till the geolocation for all the blocks are published, **whichever is later**.
- An edge case is when the drone takes off, detects and reports the geolocation of all the blocks in the arena, but crashes shortly after. Even in this scenario, the end time would be considered 250 since the drone never landed, hence assumed to land at time infinity, which is 250.

> NOTE:
>
> - In case of any disputes/discrepancies, e-Yantra's decision is final and binding
> - e-Yantra reserves the right to change any or all of the above rules as we deem fit
> - Any change in the rules will be highlighted on the website and notified to the participating teams

Important Notes

- As per e-Yantra policy and NDA, all your codes, solutions, and documents etc. are solely the property of e-Yantra (IIT Bombay)
- After completion of all tasks, teams will be selected as finalists based on their cumulative scores across all the tasks. Complete rules and instructions for the finals at IIT Bombay will be sent to those teams that qualify for the finals
- In case of any disputes/ discrepancies, e-Yantra's decision is final and binding. e-Yantra reserves the rights to change any or all of the above rules as we deem fit. Any change in rules will be high lighted on the website and notified to the participating teams

Happy Learning and All the best!!

6 Likes

Task 4B: Detect colored object from the drone camera and find its pixel co-ordinates

---

Saail Closed 3 January 9, 2023, 11:01am

Saail Listed 4 January 9, 2023, 11:29am

SD Theme: Regarding Stage2 and Rulebook