Strawberry Stacker Rulebook

blogpost-style, rulebook

Smit e-Yantra Staff Oct '21

Strawberry Stacker

1. Introduction

1.1 Background

Unmanned Aerial Vehicles(UAVs) and the Unmanned Aerial Systems(UAS) they are a part of are a large and active area of engineering, research, development and deployment.

UAVs also colloquially called drones are self-powered flying vehicles without a huSkip to main content they may have varying degrees of autonomy from fully

autonomous to fully controlled by a offboard human pilot.

A UAS is the larger system that the UAV is part of, and it includes elements such as ground stations and the data links between UAVs and ground stations; aspects which are equally critical for successful missions.

We shall be developing and deploying our application in a simulator, this is to make the most efficient use of our development time and effort. Physically flying a UAV requires a lot of time, effort in terms of setup, maintenance, monitoring and daily upkeep. The procurement and operational costs are also not insignificant.

Thankfully, the excellent open source software pipeline typically deployed on UAVs allows us to replace a physical UAV with one in the simulator, with the rest of the system remaining the same. This process is called Software-in-the-Loop (SITL) simulation and is a quick, cheap, reliable and hassle-free way to get started with developing applications. A vast majority of application development work should and does happen with such SITL simulations.

1.2 Problem Statement

Our problem statement consists of a team of multiple autonomous drones, tasked with identifying and picking boxes from in-between the strawberry rows and then stacking them in a palletized form onto two parked flatbed trucks.

1.3 Tools Used

For solving our problem statement we shall use PX4 Autopilot ecosystem for controlling the UAV, Gazebo simulator, a robotics simulator, where the simulated farm and UAV’s will reside, and ROS for integrating various aspects of autonomy required in the solution.

2. Theme Description

This theme is an abstraction of a strawberry farm. It is the harvesting scenario in the strawberry farm where the harvesters are plucking the strawberry selectively which are ready for harvesting and packing them into small boxes. The harvesters usually separate the premium quality strawberries and pack them in a RED coloured box whereas the standard quality ones are packed in a BLUE coloured box. The boxes then have to be placed on respective trucks (red boxes on red truck and blue boxes on blue truck) to ship them further. This entire task from plucking to shipping has to be done as quickly as possible.

Skip to main content

2.1. Terms

2.1.1. *eDrone*

- The *eDrone* is the Unmanned Aerial Vehicle (UAV) which is the delivery agent in the theme
- The *eDrone* is a quadcopter model designed in the Gazebo simulation environment

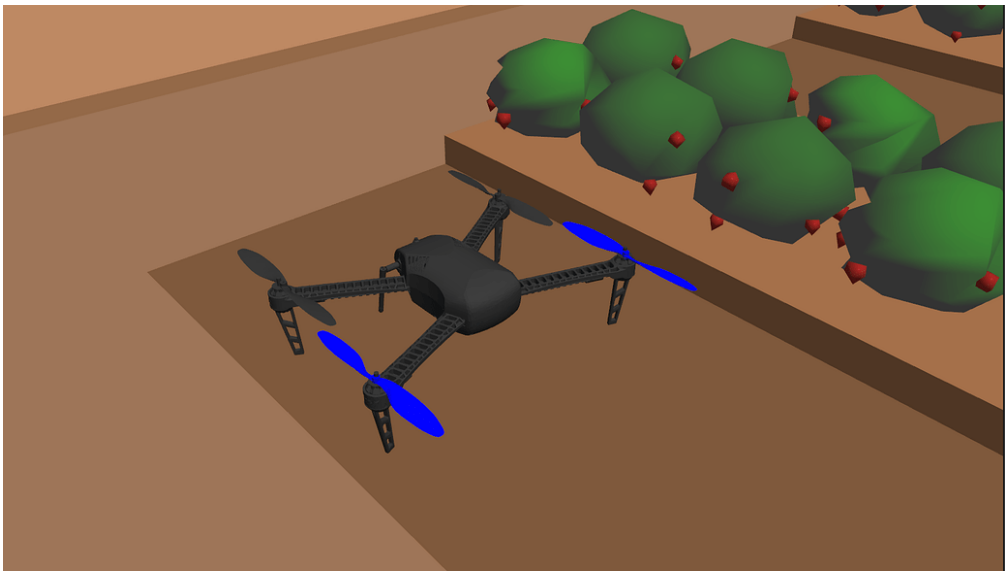

Figure 1: eDrone model in Gazebo

2.1.2. Strawberry Box

- There are two kinds of strawberry boxes being packed, the premium quality ones packed in a RED box and standard quality ones in BLUE box
- The boxes have to be transported from one place to another by *eDrones*
- Each box is a cuboid in the Gazebo simulation environment of the dimensions: L = 0.3m W = 0.24m H = 0.104m
- The boxes spawn in the gazebo environment as and when the harvesters complete packing them. Harvesters start harvesting parallelly in rows and proceed further in the same direction. As a result, the boxes can spawn simultaneously in the parallel rows. However, in any given row, boxes will spawn in a strictly increasing order by distance over the length of the row. For eg. In a row, the last box spawned is at a distance of 5m from the start of the row, then the next box will strictly spawn at a distance greater than 5m. This simulates the harvester proceeding further in a row
- Each strawberry box has an ArUco marker pasted on it by the harvesters for the *eDrones* to identify them and locate them precisely from a height. The red and blue boxes will be 1 and 2 respectively

Skip to main content

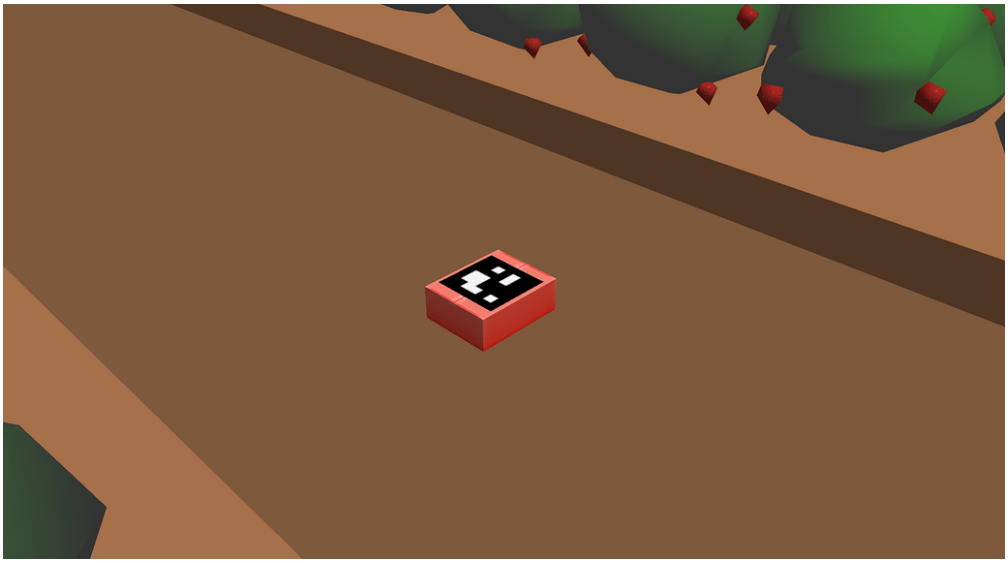

Figure 2a: Red Strawberry Box

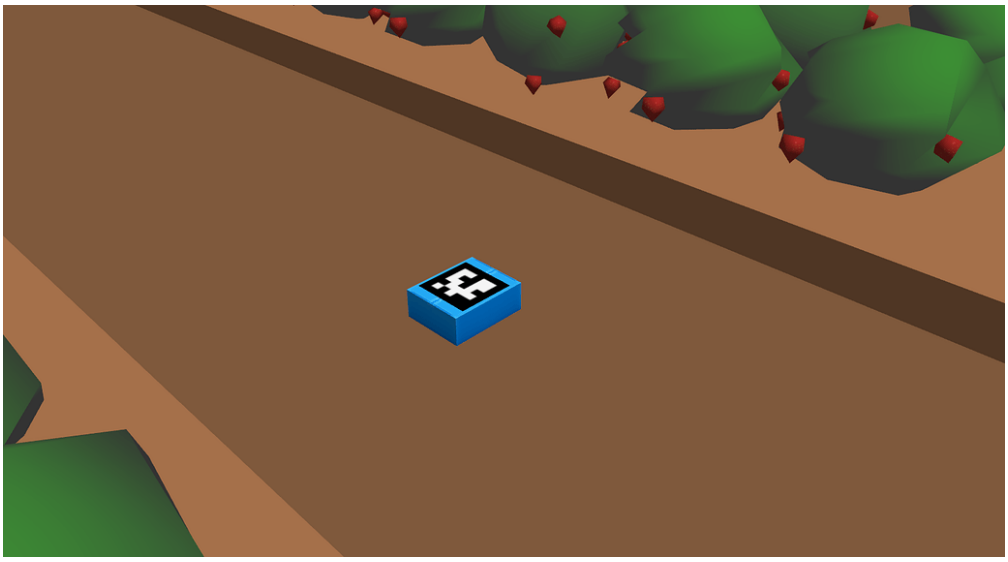

Figure 2b: Blue Strawberry Box

2.2 Inputs

- As soon as a box is spawned in a row, a message will be published on a *rostopic* named `/spawn_info` which will just contain the row number in which the box has been spawned
- The exact location and the type of the box spawned will NOT be shared but can be approximated by simple logic. For eg. the co-ordinate of a particular row is known, one can keep a track of number of boxes spawned in each row, so as soon as a new box is spawned, the approximate distance of the box can be estimated and then the *eDrone* can scan the approximate area and get the precise location of the box by searching for an ArUco marker and pick it up. The color of the box can be identified by the ArUco Id

Skip to main content

3. Arena

3.1. Strawberry Farm

- The entire arena will be a strawberry farm constructed in a Gazebo simulation environment
- The farm contains 15 rows of strawberry plants spaced at a distance of 4m. Each row of plant is followed by empty space where boxes will be spawned
- The rows are numbered from 1-15 from left to right, row 1 being the 1st row. The exact coordinate of the 1st row will be provided before the run
- The farm has two flatbed trucks which is the delivery location for all the strawberry boxes

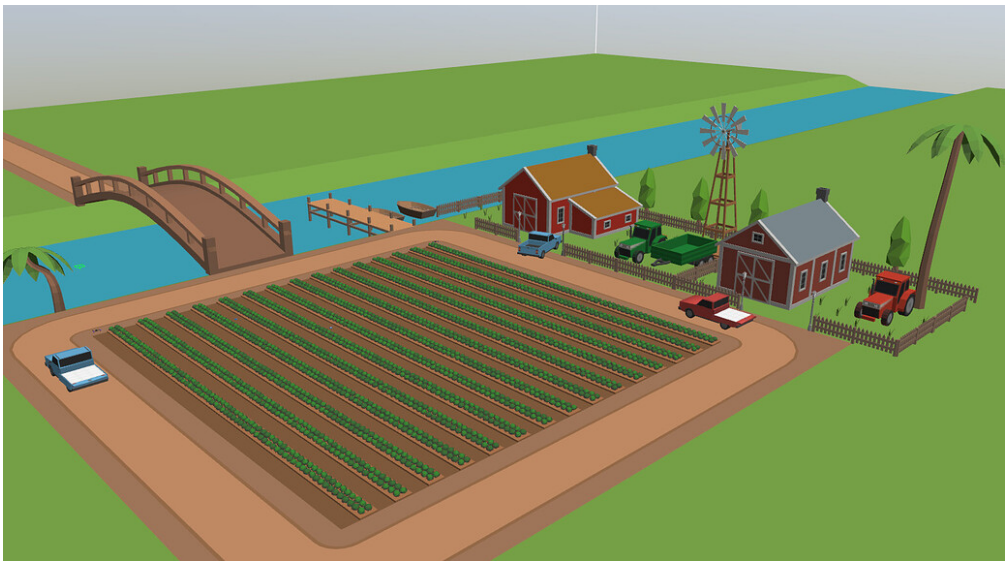

Figure 3: Farm

3.2. Delivery Truck

- A delivery truck is the area where the *eDrone* is supposed to land and drop the strawberry box.
- There are 2 types of delivery trucks, one is RED coloured and another is a BLUE coloured. Each one has to carry appropriate coloured boxes.
- The trucks have a grid drawn on them and the respective strawberry boxes are to be placed in the rectangular cells inside the grid. However, the *eDrones* can stack multiple boxes on a single cell.
- The exact coordinate of the 1st cell (which is the top left cell) as well as the length and width of each cell will be provided before the run

Skip to main content

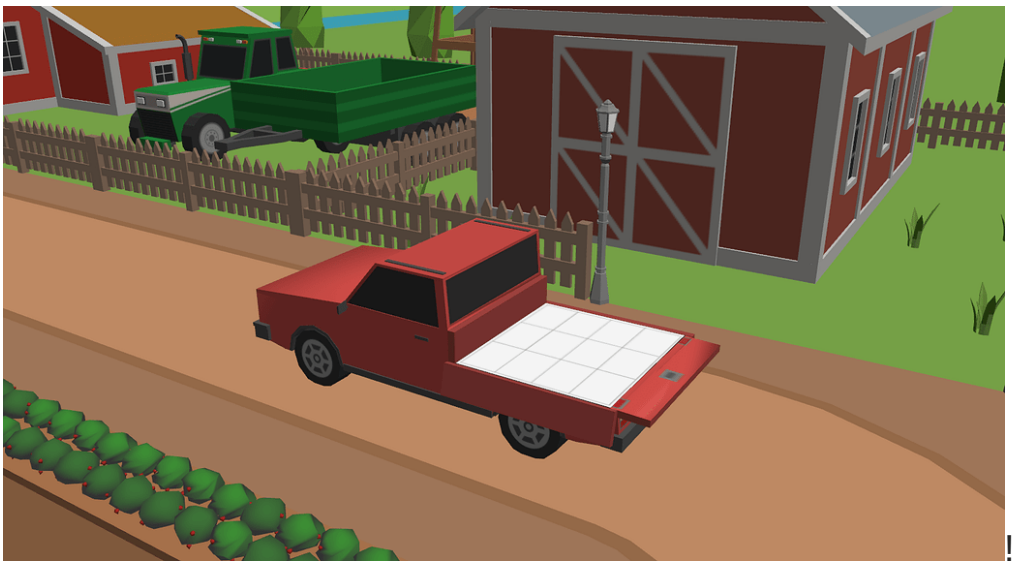
!

Figure 4a: Red Truck

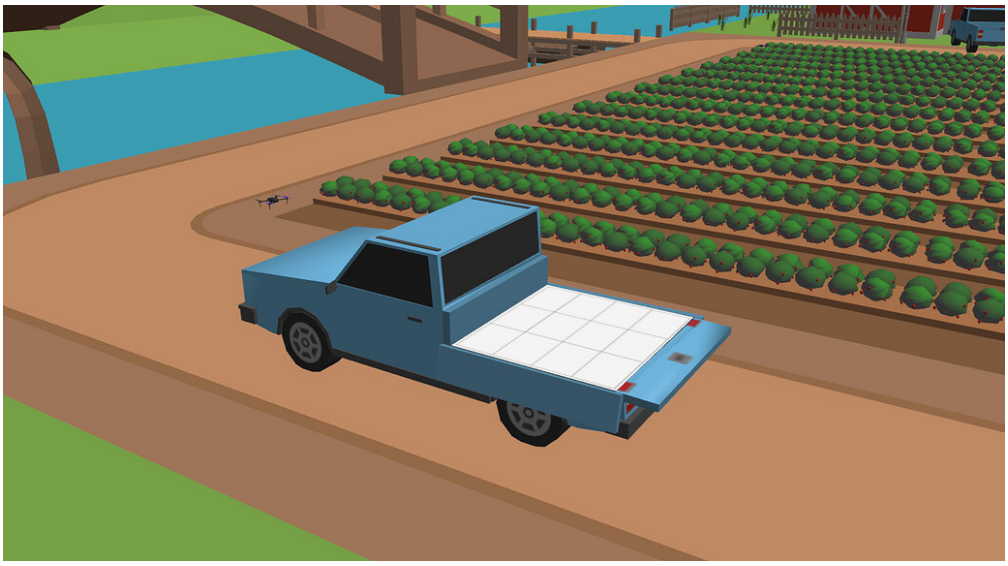

Figure 4b: Blue Truck

4. Software Specifications

4.1. Ubuntu 20.04

- The operating system on which all the softwares run is Ubuntu 20.04. Since the theme is run entirely in Gazebo 11 simulation, Ubuntu 20.04 is the only supported operating system

4.2. ROS Noetic

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

Skip to main content

4.3. Gazebo 11

- The entire theme is to be implemented inside the Gazebo simulation environment
- The official version supported in the theme is Gazebo 11.xx.
- Gazebo 11 is tightly integrated with ROS Noetic and so it comes pre-installed when ros-noetic-desktop-full is installed

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 preinstalled 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 written approval from e-Yantra

5. Theme Flow

- Participants will be given a launch file that will load the gazebo world containing 2 drones. Participants need to prepare another launch file that contains all the scripts and *rosnodes* to perform the task and launch this file before anything else. Then participants can launch the file provided which will start the gazebo simulator. The start time will be considered when this launch file is run.

> “ NOTE: The simulation time will be used throughout the theme to maintain consistency across all teams

- After launching the scripts/*rosnodes*, the drones should takeoff and start their delivery jobs. The spawning of the strawberry boxes will be in a pseudo random order and their location will be shared on the *rostopic* as soon as it spawns
- The *eDrones* re allowed to pick any strawberry box in any order, there is no specific requirement to follow any kind of first come first serve pattern
- The *eDrones* have to pick and place as many number of boxes in 5 minutes

Skip to main content

6. Theme Rules

6.1. Theme Rules

1. The run will be of finite time of 5 minutes ie. 300 seconds
2. The *eDrones* have to successfully complete as many jobs as possible in this time
3. A job involves picking and placing of a strawberry box

6.2 Picking Rules

1. The *eDrone* can only pick one box at a time

6.3 Delivery Rules

1. The strawberry box should be within the area over the truck. Any strawberry box fallen from the truck will not be considered as delivered.

2. The *eDrone* should not be more than 1m above the truck before releasing the box

6.4 Re-positioning Rules

1. Any kind of re-positioning (resetting the model poses or the simulator) is **NOT** allowed during the run
2. If the *eDrone* 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

- Successfully picking and placing a strawberry box according to the theme rules constitutes a successful delivery.
- Number of **strawberry boxes delivered** during the run is termed ***SBD***
- Each successful delivery earns a base rate of ₹10. Abbreviated as ***BR***
- Hence the total amount that can be earned during the run is as follows:

$$Total\ ₹ = \Sigma\ SBD * BR$$

Skip to main content

> - 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!!

Cannot download rulebook

SS: Task 6 Announcement

Task 6 mark regarding?