Task 3.2 - Survey & Rescue

1 Problem Statement

- The Drone should start at [0, 0, *z*] w.r.t. to WhyCon frame, where *z* is your ground value.
- The drone should take off and visit all the 36 cells (in order from A1 to A6, then B1 to B6, ...F1 to F6) on the arena and hover at a distance of 1½feet from the ground for 5 seconds over each cell and then land on the ground the last cell ie F6, you may do this my disarming the drone.

2 Procedure

- Record the WhyCon coordinates of each of the 36 cells by utilizing the *capture_cell _coordinates.py* from the scripts folder of the updated package. Follow a demo video for recording coordinates of cells here
- This will create a *.json* file in the scripts folder in which the mappings of cells and whycon coordinates is done. You need to read this *.json* file in your python script in order to visit the cells. You can refer to the internet on how to load and use *.json* file. This link might help.
- Create a new script or use the previously developed script with little modification to visit all the 36 cell's co-ordinates from the *.json* file.
- The maximum allowed errors are ±0.5,±0.5& ± 1.0 in x, y & z axis respectively. Try to achieve as minimal error.

3 Expected Output

- Refer the video link to get an idea.

4 Points to Remember

- Please do not change any of the templates.
- Do NOT change the inner and outer diameters of whycon markers, keep it default else the grading will affect.
- The real Drone provided is not exactly same in terms of response w.r.t to the Drone in Gazebo.
- Thus, even when just throttle is being tuned, it should ideally just move in *z* axis; that is not the case. The drone tends to move in both *x* & *y* axis also. One way to tackle this problem is to tie a 20-30cm thread at the bottom of the Drone so as to:

- Keep it within the WhyCon frame while tuning.
- Protect it from crashing.

5 Submission Instructions

Follow the instructions below to submit your Task.

5.1 Bag File:

- Next, after tuning, when you want to record the bag file for submission, run the launch file name record_rosbag. This file will launch both your position_hold.py and the rosbag recording at the same time.

  ```
  roslaunch survey_and_rescue record_rosbag.launch duration:=240
  ```

  WARNING: For recording/creating the bag file, use ONLY record_rosbag launch file. Recording (using record_rosbag) should cover from the Arming of the drone till it achives its target, with this time limit.

- This will record a bag file (titled markers.bag by default) for a duration of 240 seconds of real time. This bag will be stored in the survey_and_rescue package folder in the src folder of your catkin workspace.

- Before submitting make sure to verify your bag file, by Check the number of messages, which should be in between 7190-7210, which can be done by following line

  ```
  rosbag info markers.bag
  ```

- Further verification can be done by *plotjuggler* package, to see errors while tuning or for performance verification, after the bag file has been created.
  NOTE:
  Repeated running of this launch file will overwrite the markers.bag file, please make sure you have a backup if you so wish. Alternatively, you can use the argument rec_name as follows to have a custom name

5.2 Python Script:

- You must submit your position_hold script that you developed.

- Rename the python script as SR_<team_id>_3_2.py

5.3 JSON File:

- You must submit the .json file that you recorded to complete this task.
- Rename the python script as SR_<team_id>_cell_coords.json

Store the files mentioned above into .zip file and rename the zip as <team_id>.

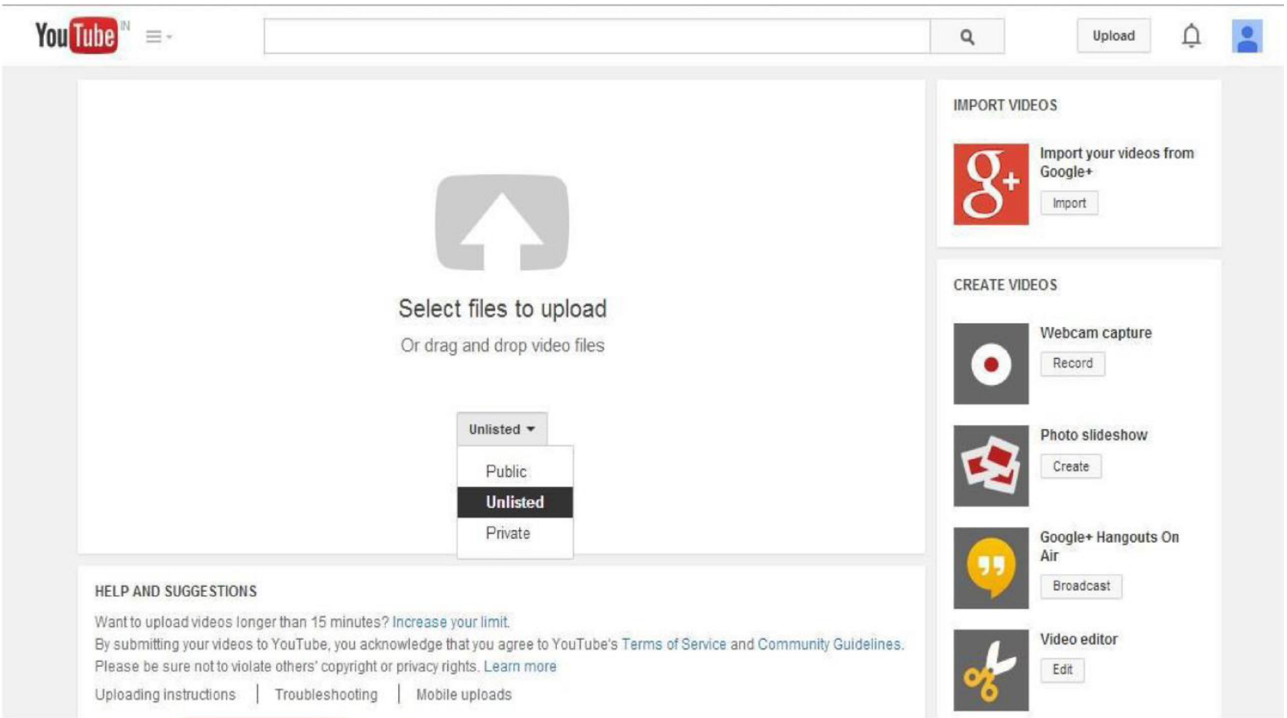
5.4 Video:

- Upon verifying that your task is complete, record a maximum 5-minute video using a screen recorder like simplescreen recorder or kazam.
- The video must be as follows:
  1. Screen Recording starts
  2. Team Slide –All member's details in a slide , included in this folder.
  3. Any One member of the team, running the necessary scripts or launch files in terminal.
  4. After the script begins, make fullscreen the whycon image_out window.
  5. Video completes when the drone reaches the last cell.
- Please refer this video link as a sample recording video.

The video should not be edited in any manner. Teams uploading an edited video will be disqualified from the competition. **e-Yantra reserves the rights to disqualify any team if any foul play is suspected.**

Uploading video/s on YouTube:

- Upload a one-shot continuous video with the title eYRC#SR#Task3_2#<TeamID>
  (For example: If your team ID is 1234 then, save it as eYRC#SR#Task3_2#1234)
- Please note that while uploading the video on YouTube select the privacy setting option as Unlisted as shown in Figure 1. You need to upload the video as instructed on the portal.



HELP AND SUGGESTIONS

Want to upload videos longer than 15 minutes? Increase your limit.
By submitting your videos to YouTube, you acknowledge that you agree to YouTube's Terms of Service and Community Guidelines.
Please be sure not to violate others' copyright or privacy rights. Learn more

Figure 1: Uploading unlisted video on YouTube

NOTE: You must upload all of the following files:

- bag file
- Python code

Please place all these files inside a .zip file before uploading. You must also upload the video to YouTube with the instructions above. You will have to submit the video link on the portal.

Please follow the naming convention strictly as specified in each step. Failure to do so may lead to repercussions. The deadline for Task_3.2 is 8th January, 2020.

Your final .zip output must be of the following structure:
<team_id>_3_2.zip

- SR_<team_id>_3_2.bag
- SR_<team_id>_3_2.py

Instructions for uploading the folder will be provided on portal.

Good Luck!!!