Explorer Bot

1. Introduction:

Robots need not know, but we as creators of these machines need to reason out why we send robots to other planets. The reason is curiosity. Humans are curious to explore the unknown. Scientists and Technologists are ever ready to go beyond the known theories to reason out and solve mysteries. Space exploration is one such mystery. Considering any planet in our solar system, we ponder on questions such as, what is the temperature? What is the quality of soil and rocks? Are the conditions suitable for sustaining life forms? And so on…. Finding answers to these questions gives us an opportunity to push the boundaries of current technology. These findings eventually lead to the growth of human civilization and inspire scientists and technologists towards more discoveries and inventions.

Inspired by India's recent achievements in the "Space exploration" domain, in this edition of e-Yantra Robotics Competition (eYRC-2016), we present a theme *"Explorer Bot"* where you build a robot, missioned to find meaningful **Substances** in a recently discovered e-Planet 903 and send their coordinates to the **Base Station**. Through this theme, we will help you to discover your talent in designing and building a bot from scratch. We have divided the theme into a number of "Tasks" to build the robot in a step-by-step manner making the process easier and interesting for you. You will learn many new things like microcontroller programming, sensor interfacing, robot construction, control systems and controlling the robot using the Robot Operating System (ROS).

The robot has to travel through an arena, representing a terrain, avoiding **Obstacles**. On the arena, the substances are represented by markers. Robot is equipped with an **on-board camera**, an **inertial measurement unit** and a **wireless communication** module. PC/Laptop running the Robot Operating System (ROS) represents the **Base Station.** The ROS processes the given map of the terrain (assume that this map is provided by another robot which was missioned to navigate the area and design its map) and assists the robot to navigate through the terrain.

While exploring the terrain, the robot finds samples of Substances (for example soil, rock, and water). Robot has to collect information such as the type of Substance and coordinates of the location of the Substance. The robot communicates the information gathered during the exploration to the Base Station.

The challenge is to complete this task in the shortest time possible, find the correct Substance and send the information to the Base Station. The robots that perform the task in accordance with the rules set for this task will be declared the WINNER.

2. Theme Description:

- **Explorer Bot (EB):** This is the robot that you build which is capable of navigating the Arena, communicating with **Base Station** (defined below) and processing images captured through a camera on-board.
- **Base Station:** A PC/Laptop with **Robot Operating System (ROS)** environment represents the Base Station. The communication between EB and Base Station is done wirelessly through XBee.
- The arena consists of a 7x7 grid. Each cell in the grid is termed as **Site.** In the middle of each Site, is a square referred to as **Object Point**. Refer to Figure 1.
  - **Site**: As shown in Figure 1, each Site is referenced by a XY co-ordinate system using the numbers along the X and Y axes. The co-ordinates (2, 3) refer to the Site as marked in Figure 1.
  - **Start**: The site with co-ordinates (0, 0) is termed as Start. The **Explorer Robot (EB)**, the Robot designed by you will start from this point for exploration.

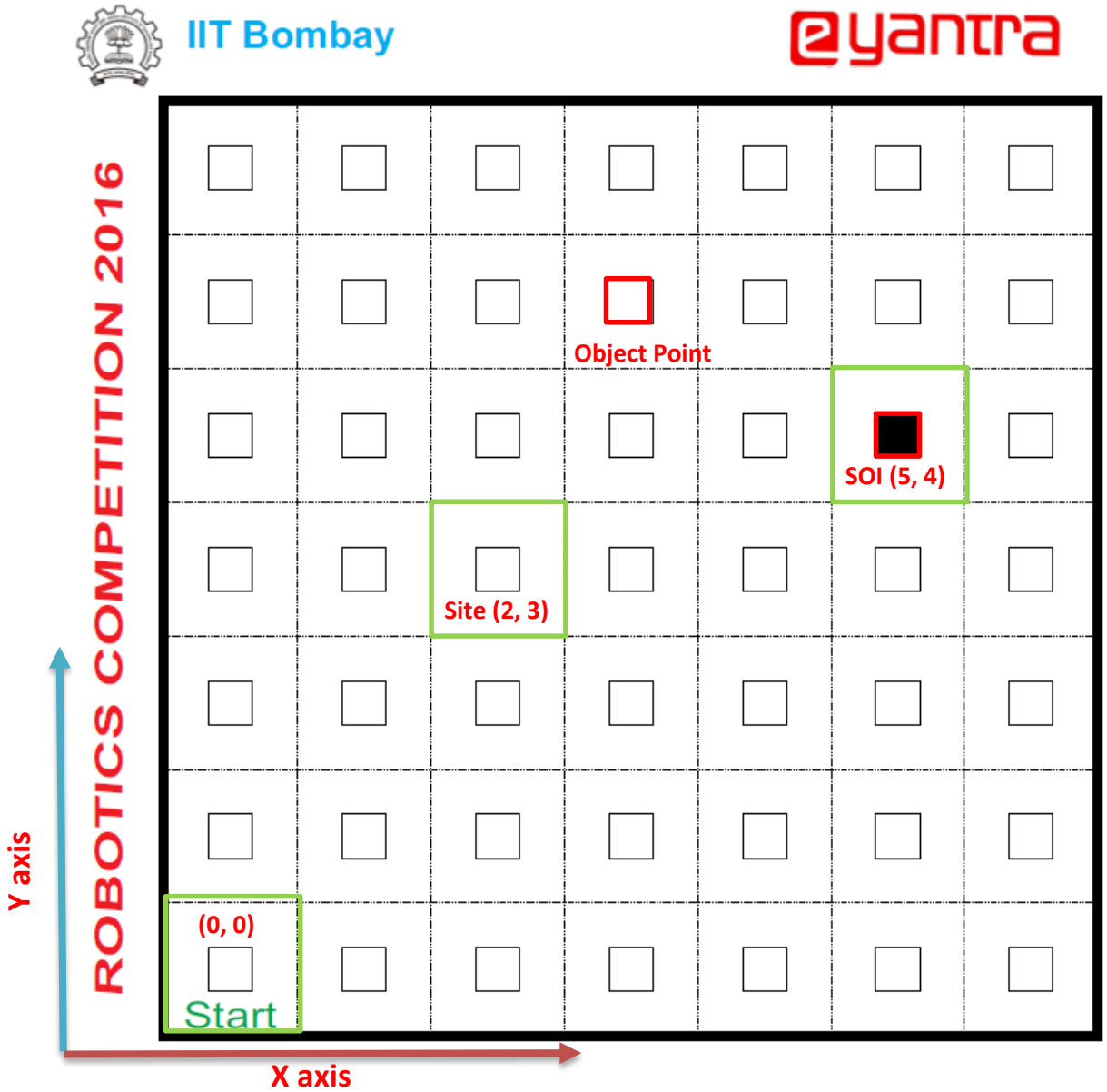

Figure 1: Arena with Defined Terms

Object: An Object is represented by a block with dimensions **10x10x10** cm with **ArUco marker** **(Refer Task 3.2)** pasted on one of its lateral faces. Preparation of ArUco Markers is explained below. The Object can be placed in any of the Object Points except Start. Object can represent a **Substance** or an **Obstacle** depending upon the ID of the ArUco marker. There are a **maximum of eight unique Objects** used in this theme.

- **Substance**: Object is considered as Substance if the ID of its ArUco marker is present in **ArUco Table**. ArUco Table is given in **Table 1**. Note that the ArUco IDs in the range 200-214 represent Substances.
- **Obstacle:** Object is considered as Obstacle if the ID is not present in the **ArUco Table**. Note that ArUco IDs other than 200-214 represent Obstacles.

ID Number	Characteristic
200	Sandy
201	Clay
202	Silty
203	Rain Water
204	Ice
205	Snow Water
206	Sedimentary
207	Metamorphic
208	Igneous
209	Oxides
210	Carbonates
211	Phosphates
212	Sulfides
213	Native Element
214	Silicates

Table 1: ArUco Table

- **Site of Interest (SOI):** The Site in which the **Object is present** is called Site of Interest. The SOI can be any of the 49 Sites except Start. There will be a maximum of eight SOI in this theme. SOI (5,4) is shown in Figure 1

In this theme, **EB** starts navigation from **Start**. Upon visiting each **SOI**, EB has to detect the **ArUco ID** using camera and send this to the*99 **Base Station**. Depending upon the ArUco ID, Base Station classifies the **Object** as **Substance** or **Obstacle**.

A. INPUTS:

1. **SOI List:**
   a. SOI list consists eight elements corresponding to each Object. Each element has three parameters. Each element is separated by comma and each parameter is separated by colon.
   b. SOI List consists of three parameters for each Object to be placed in the Arena: (i) Site co-ordinates (ii) ArUco marker of the Object and (iii) Orientation of ArUco marker.
   c. Orientation of the ArUco marker is denoted by ‘x’ and ‘y’.

- If orientation is ‘x’ then the ArUco marker faces towards X-axis.
- If orientation is ‘y’ then the ArUCo marker faces towards Y-axis.

Note that orientation of ArUco markers in the column with x-coordinate as zero can **only** be facing X axis. Similarly, the orientation of ArUco markers in the row with y-coordinate as zero can **only** be facing Y axis.

d. Let us consider an example: if SOI List is: {(2,3): 217: ‘x’, (6,4): 212: ‘y’, (5,2): 209: ‘x’, (4,6): 205: ‘x’, (1,1): 220: ‘y’, (5,0): 213: ‘y’, (1,5): 200: ‘y’, (3,4): 219: ‘x’}, then Arena looks like Figure 2. (Figure 2 illustrates the placement of Objects; placement of ArUco markers on the Objects is explained below)

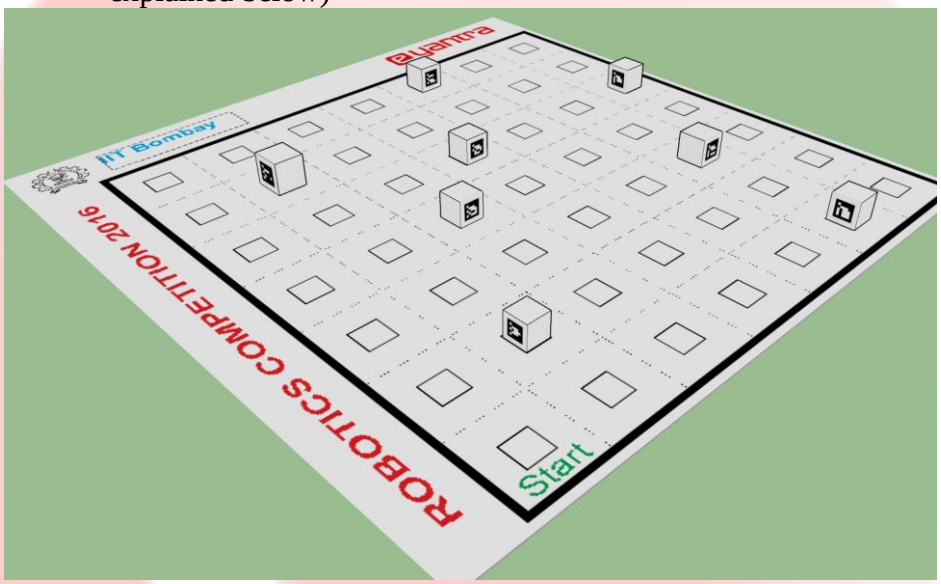

Figure 2: Placing Objects according to Input 1.

2. **RViz Map:**
   a. This is an image given to the Base Station for localization of EB and simulation of Arena. The RViz map is exact replica of the arena after setting up the Objects as specified in SOI List.
   b. Base Station processes the RViz map and gives commands to the EB for navigation. It uses ROS to co-ordinate the peripherals of EB and RViz to simulate the arena environment.
   c. With respect to the example illustrated in Figure 2, RViz map looks as shown in Figure 3.

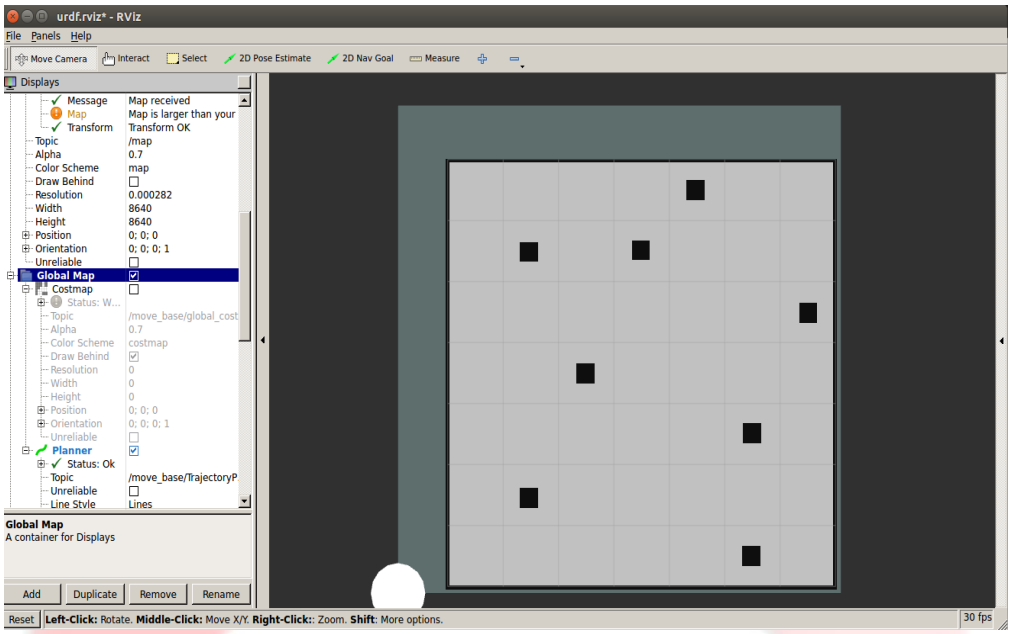

Figure 3: RViz Map

- **OUTPUTS:**

  a. The Output is a String in the following format:

  Serial Number of the Site visiting: Previous Site: Current Site: ArUco ID: Characteristic

  b. Output is displayed on the Base Station (in the terminal window of PC/Laptop).
  c. With respect to the example illustrated in Figure 2,
     i. if the EB is moving from Start to (2, 3), then the Output is:
     1: (0, 0): (2, 3): 217: OBSTACLE
     ii. if the EB is moving from (5, 2) to (4, 6), then the Output is:
     4: (5, 2): (4, 6): 205: Snow Water

In summary, this theme involves the following:

- Arena is set up using SOI List.
- RViz Map is given to the Base Station.
- Base Station analyses the RViz Map and guides EB on the Arena through commands. Simultaneously, RViz emulates Arena of the Base Station.
- After visiting each SOI, EB detects the ArUco ID present on the Object using camera.
- EB sends the detected ArUco IDs to Base Station.
- Base Station classifies and displays the Output on terminal window (in the format shown in OUTPUTS section).
- After visiting all the Sites in SOI list, EB stops. "**Mission Accomplished"** message is displayed on the terminal window of the Base Station.

3. ARENA:

The Arena for this theme is a simplified abstraction of the surface of an imaginary e-Planet 903. Each team has to prepare the Arena that consists of two major steps:

- Printing the arena design on flex sheet.
- Preparing ArUco markers and Objects.

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

A. Printing the arena design on flex sheet:

The Arena design to be printed on a flex sheet is as shown in Figure 4. A document (.pdf) file containing the Arena design will be provided to the teams. Each team prints the flex design according to the directions given in the .pdf file.

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**.

Details of Arena design:

With reference to Figure 4,

- Dimensions of the flex sheet are 8ft x 8ft.
- The arena consists of 7x7 grid made of black dotted lines. Each cell has dimension of 1ft x 1ft.
- The Object Point has dimensions of 10cm x 10cm.
- Note that the centers of Site and Object Point coincide. The center is the point where the two diagonals meet.
- Teams are not allowed to make any changes in the arena design. Any team making any modification whatsoever will be disqualified from the competition.

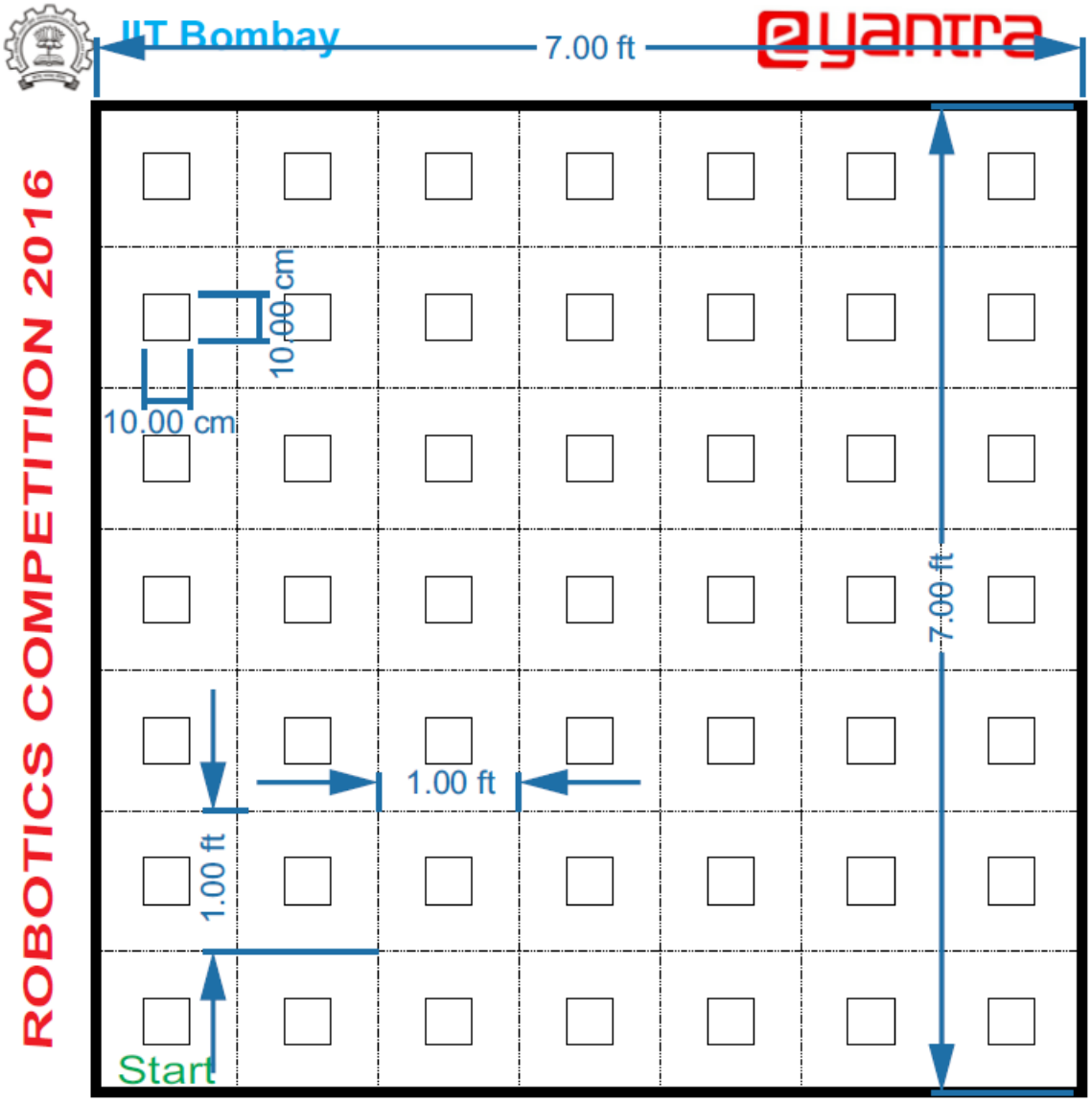

Figure 4: Dimensions of the Arena

B. Preparing ArUco markers and Objects:

- Print ArUco markers having dimensions 100mm x 100 mm: (Refer ArUco_marker_detection)
  - Marker Size =40mm
  - Marker Padding = 30mm
- All the Objects are of similar shape i.e., a cube of dimensions 10cm x 10cm x 10cm.
- Prepare eight Objects using Thermocol sheets. (If the thermocol sheets of required dimensions are not available then the teams may cut or join the available sheets on their own).
- Paste the ArUco markers on one of the four lateral faces of each Object. The other faces of the Object have to be left as they are. A sample Object is shown in Figure 5.

- You may mark on the top side of Object to identify the orientation of ArUco marker such that the ArUco marker pattern is in correct orientation (Note that if the Object is placed upside down then the detected ArUco orientation will be considered wrong).

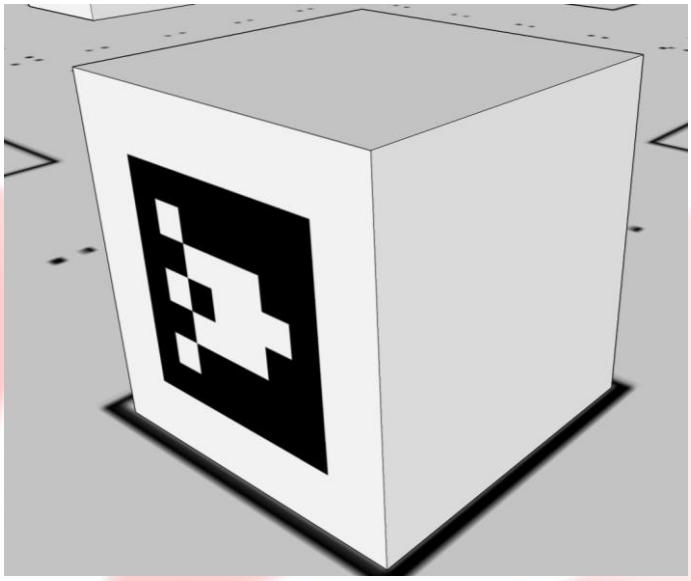

Figure 5: Sample Image of Object

C. Setting up the arena:

According to the SOI list, place all the eight Objects on the Arena.

Note: If the arena is found damaged or in a condition that makes evaluation difficult or not as per the given SOI list, e-Yantra has the right to disqualify a team. The final decision is at the discretion of the reviewer.

4. HARDWARE SPECIFICATIONS:

A. Use of Components:

- All the participating teams must use only the components which were sent to them in the kit. Only one set of components given in the kit is allowed per team.
- The robot should be completely autonomous. The team is not allowed to use any wireless remote or any other communication protocol for manual control of the robot.
- Teams are allowed to create any type of mechanical mount for mounting camera.

B. Power supply:

- The robot can be powered through Li-Po battery or auxiliary power supply. These are supplied with the kit.
- The team cannot use any other power source for powering the robot.
- The team can use auxiliary power source for powering the robot during testing but the final demonstration should only be made using the battery powered robot.

Note: No other expansion and/or microcontroller-based board shall be attached to Raspberry Pi and ATmega2560.

5. Software specifications:

- e-Yantra has provided all teams with an e-Yantra's Version of the **ros_berry_customized.img** that was already given to the teams with SD card in Task-2. This image has all the software and drivers pre-installed.
- The teams must use Python to write their code.
- You are allowed to use only inbuilt Python Libraries. Use of any other external libraries is not allowed and will result in disqualification.
- As per e-Yantra policy, all your code and documents are open-source and maybe published on the e-Yantra website.

6. Theme Rules:

- The maximum time allotted to complete the task is 10 minutes. A maximum of two runs will be given to a team (the best score of the two runs will be considered as the team's final score). A maximum of two Repositions is allowed in each run.
- The team should place the EB at the Start with castor wheel pointing in X axis and switch it ON when asked to do so by reviewer.
- Then the team should do all the required initializations at the Base Station and be ready to launch the final script.
- When the team runs the final script, the timer is started.
- Once the robot starts, human intervention is **NOT allowed**. If made, the intervention is treated as a **Reposition**. (Reposition Rules are discussed below).
- The EB can follow any order to visit the Sites given in the SOI List.
- SOI is marked as **visited** only if
    - EB moves towards the Object with the camera facing the marker **AND**
    - Output gets displayed at the Base Station irrespective of the correctness of classification.
- The score will be regarded as valid only, if the EB visits at least **two** SOI mentioned in SOI list.
- SOI is marked as **correctly classified** if output at the Base Station is shown according to SOI list and ArUco Table.
- Each time EB touches any Object on the Arena it is counted as a **Penalty**.
- Team will be awarded a **Bonus Point** if the EB visits and detects all the Sites mentioned in SOI list correctly according to ArUco Table without any penalty.
- The end of the run is indicated by a message on Base Station terminal with text **"Mission Accomplished"** and the timer is stopped.
- The score of both the runs will be recorded independently and **best of two** runs will be considered as the team's final score.
- Teams are not allowed to keep anything else inside the arena other than the robot. The time measured by the reviewer will be final and will be used for scoring the teams.
- The robot is not allowed to make any marks while traversing the arena. Any robot found damaging the arena will be immediately stopped and **Reposition** will be allowed as per rules.

Reposition Rules:

- Reposition can be done if the EB:
  - is found to be displacing any Object or damaging the Arena
  - leaves the Arena
  - gets stuck somewhere in the Arena.
- For a Reposition the EB should be placed at last correctly classified Object with camera facing the ArUco marker.
- If the all previous ArUco marker detected incorrectly, EB should be place at the start position of Arena by placing castor wheel pointing in X direction.
- Each Team is allowed a maximum of two repositions in each run. Both repositions require the approval of reviewer, the team is disqualified if the robot is handled within the Arena without approval.
- During Reposition the timer is not stopped.
- If the team runs out of number of Reposition then the run is stopped and the score will be calculated until that point of time. In this case time will be considered as maximum.

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 highlighted on the website and notified to the participating teams.

Judging and Scoring System:

- The timer for a team starts at the moment the final script is run.
- The timer will stop as soon as the robot displays "Mission Accomplished" at the Base Station.
- The best score of the two runs will be considered as the final score of the team.
- The total score will be calculated by the following formula:

Total Score = (600-T) + (VS * 150) + (CC * 100) - P + B

Where:

- **T (Total time):**
  It is the total time in seconds to complete the task.
- **VS (Visited SOI):**
  - SOI is marked as **visited** only if
    - EB moves towards the Object with the camera facing the marker **AND**
    - Output gets displayed at the Base Station irrespective of the correctness of classification.
- **CC (Correct Classification):**
  It is the number of Objects classified correctly as mentioned in the SOI list according to the ArUco Table.
- **P (Penalty) :**
  Penalty of 50 points is imposed each time the robot collides with any Object or leaves the Arena.
- **B (Bonus) :**
  A bonus of 80 points is awarded if the EB visits and detects all the Sites mentioned in SOI list correctly according to ArUco Table without any penalty.

ALL THE BEST!!!!!