Pharma Bot Rulebook

shyama 1 December 22, 2022, 6:50am

Pharma Bot Rulebook

1. Introduction

Tackling challenges in mobility is of the utmost importance for a city to become “Smart”. Many industries began using home-delivery through the years as a way to enhance their services. However, this technique necessitates the use of labour as well as a vehicle, which increases fuel consumption. What if we sent robots to your door to bring medicines? Just consider the amount of fuel we will be able to save and the decrease in pollution. Robots can effortlessly move through any place, while conventional delivery vehicles often get trapped in traffic.

As a part of this year’s e-Yantra Robotics Competition (eYRC 2022-23) theme, ***Pharma Bot (PB)***, we attempt to address this issue by building autonomous delivery robots that “pick up” medicine packets from pharmacies and distribute them across our “Smart city” arena.

The premise for the theme is a city in the future with a specialised grid set aside for medical shops. The robot must navigate the arena, pick up the necessary packet from a particular shop, and transport it to the right location after being given the customers’ requests for various medical packets that need to be delivered. The robot must take into account a variety of arena parameters, including traffic signals and impassable roads from ongoing work.

The major challenges include **Image Processing, Line following using Image Processing, Algorithm Building, Robotic Navigation and Localization, Robotic Simulation, Python, and Lua programming.**

Time is a key factor, and the aim is to finish the assigned tasks as quickly as you can. The teams who complete the theme implementation the best while adhering to the rules will be declared the competition’s WINNERS.

ALL THE BEST!!

2. Theme Description

The Pharma Bot theme aims at automating the delivery of medicines in a city arena. The robot must collect various kinds of information from a given configuration image and should deliver the medicine packages to the correct target locations.

3. Arena Description

The arena of this theme is a **5x5 Grid** made up of **25 Cells**. This Grid structure represents the map of a smart city and is divided into two parts.

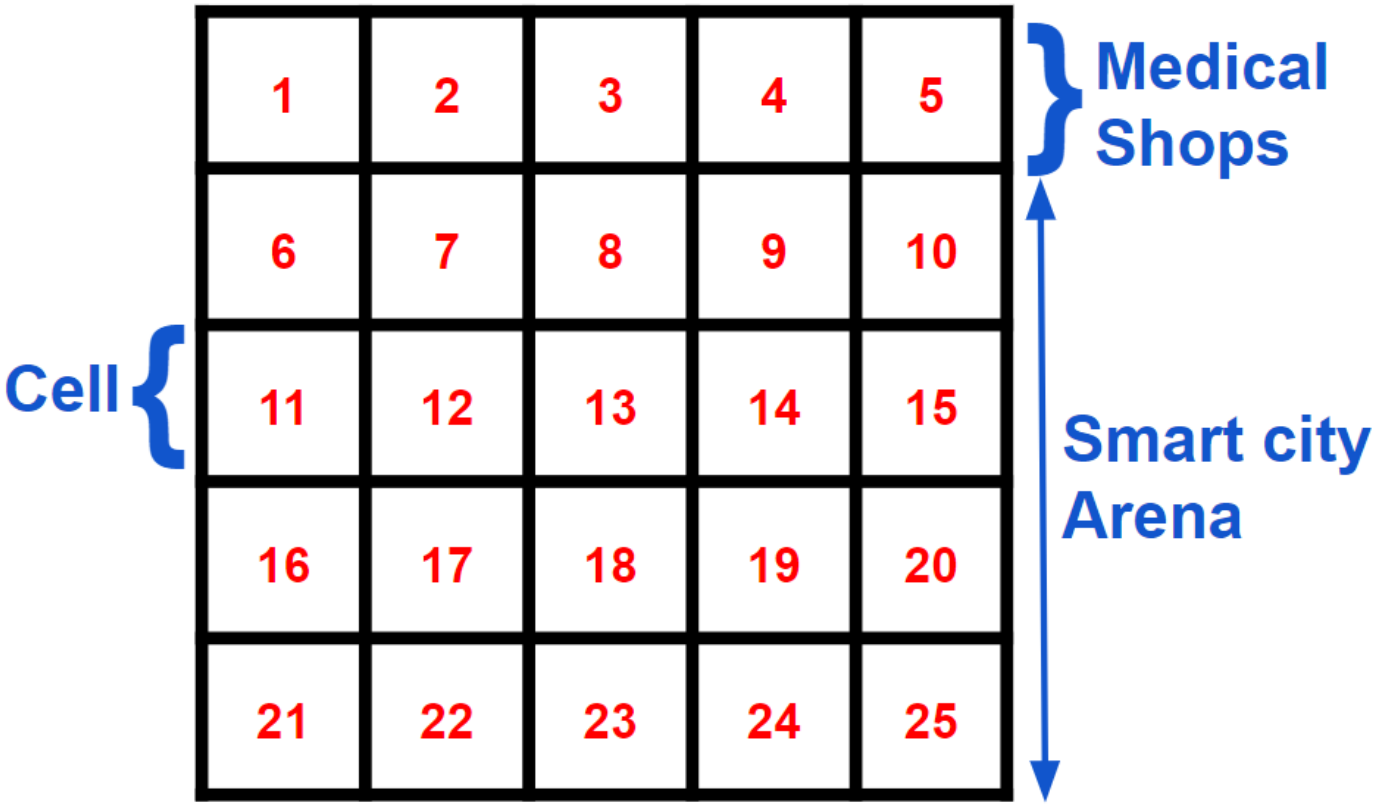

Figure 1: Grid numbering

A. Medical Shops:

- Cell numbers **1 to 5** represent five medical shops (**Cell 1 = Shop_1, Cell 2 = Shop_2** and so on). The medicine packages to be delivered across the city are stored in these shops.

B. Smart city Arena:

- Cell numbers **6 to 25** represent the **Smart city** arena. The medicine packages should be delivered to these Cells by the robot.

3.1 Input Images

The test images will be given to teams for each run.

- **Test image:** This image will be used in order to extract the following information: **Start node, Horizontal roads under construction, Vertical roads under construction, Location of traffic signals, Details of medicine packages (colour and shape) kept in the medical shops, and End node**. The details should be extracted from these images using OpenCV and Python. The extracted information must then be used to **set up the arena in the CoppeliaSim** simulator. (Details regarding the virtual arena are explained in the next section)

A sample test image is shown in Figure 2.

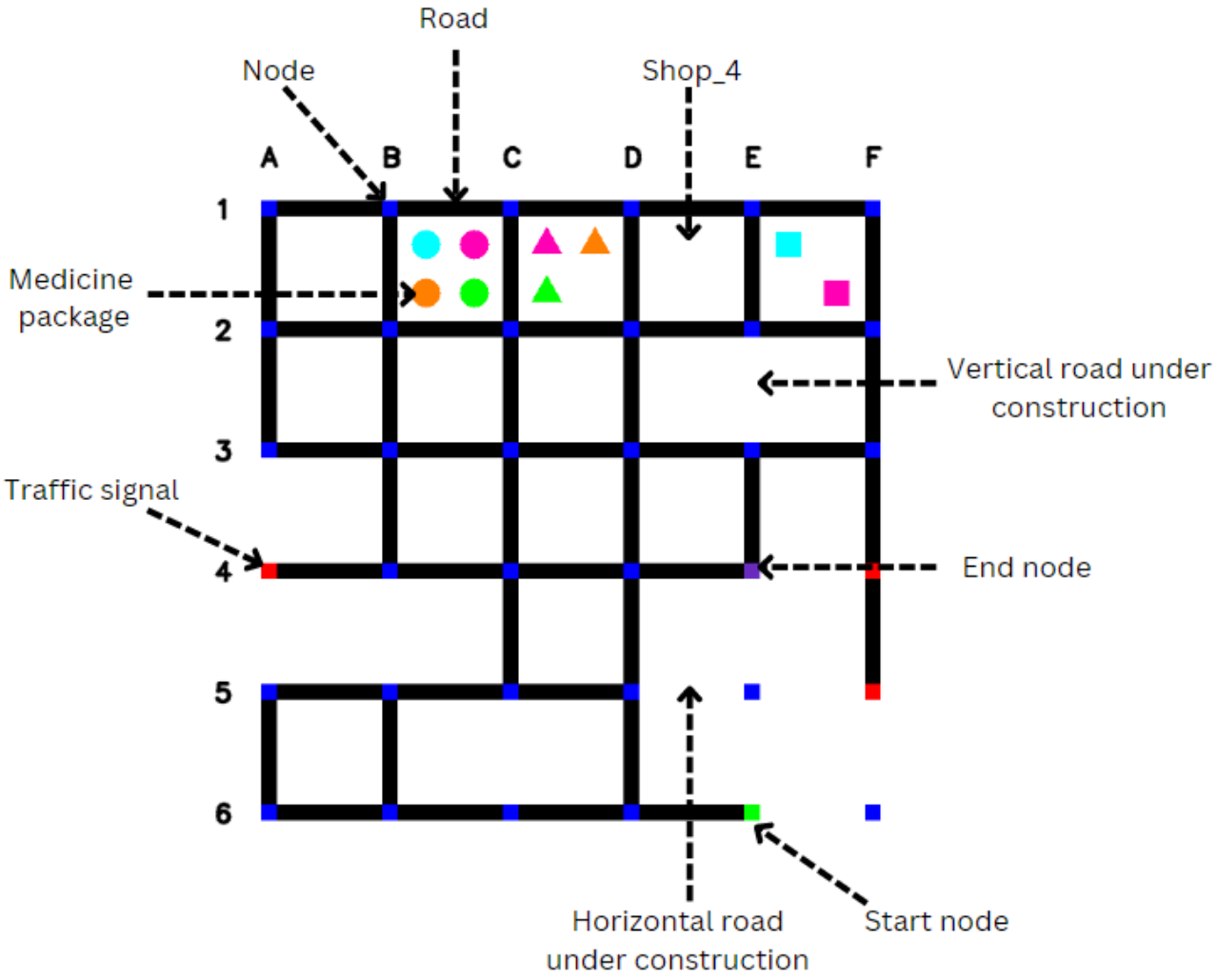

Figure 2: Sample Test image

The description for each of the parameters given in the test image are as follows:

A. Nodes

- The **point of intersection** between a vertical and horizontal road is called a Node. The small **dark blue squares** visible in the sample image are the nodes.
- There are six columns named **A,B,C,D,E,F** and six rows numbered from **1,2,3,4,5,6**. Each node is named using the column and row intersection.

  1. Start Node: The **Green square** shown in Figure 2 (**E6**), is the start node. The robot should start its traversal from this node ONLY. The Start Node can be present in any column but ***row number will remain 6***. The robot should always face in the upward direction (i.e robot's front should be facing the title printed on the top of the arena) where the medical shops are present.

  2. End Node: The **Purple square** shown in Figure 2 (**E4**), is the end node. The robot should complete the package deliveries required and stop at this node. The End Node can be present in any column but ***row number will be 3, 4 or 5.***

B. Roads Under Construction

- The **black lines** by which the grid is made up of are called **Roads**. Some of the roads in the arena are under construction. This means that the ***robot will NOT be able to traverse through these roads.***
- Roads are named using the nodes between which the road is present.

1. **Horizontal Roads Under Construction**: These are horizontal roads which are **missing**. For example, the horizontal road under construction marked in Figure 2 will be represented as **"D5-E5"**.
2. **Vertical Roads Under Construction**: These are vertical roads which are **missing.** For example, the vertical road under construction marked in Figure 2 will be represented as **"E2-E3"**.

C. Traffic Signals

- Traffic signals are represented using **Red squares** in the test image. Each signal is represented using the node at which it is present. For example, the traffic signal marked in Figure 2 will be named **"A4"**.
- If the robot encounters a traffic signal, it will have to **wait there for 5 seconds (in real-time)** before proceeding further.
- Failing to stop at a traffic signal will cause a penalty. Details regarding penalty can be found in the Judging and Scoring section.

D. Pick-up Nodes(PN)

- Each Medical Shop has a Pick-up Node (PN), from where the Packages of that Shop can be picked up as shown in Figure 3.

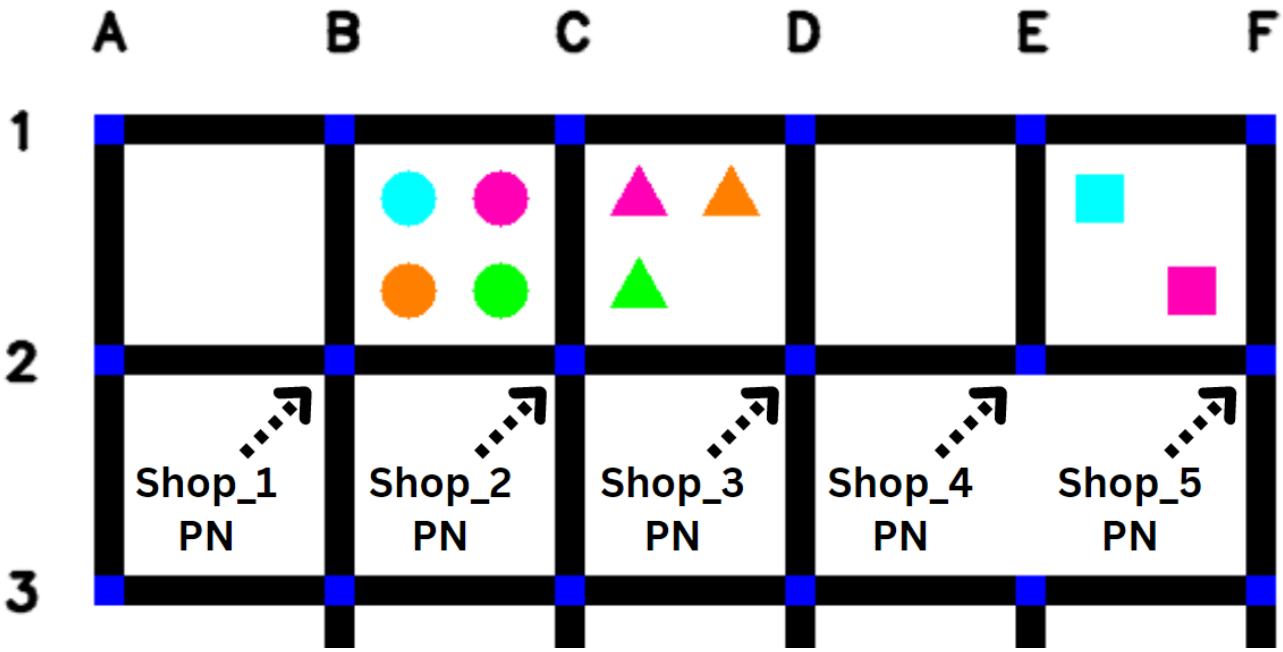

Figure 3: Pick-up Nodes

- Note: For this theme, teams **DO NOT have to create any mechanical structure (robotic arm)** for picking up Packages as the robot does not actually pick up any Package. It only needs to travel to the PN and indicate the Pick-up of a Package. **Pick-up and Drop-off** of packages using **RGB LEDs** are discussed in the Theme Run section below.

- Packages can be picked up from the shops in any order and can be dropped off to the required location in any order - the ONLY constraint being that **at any given point of time, the robot can carry a maximum of 3 Packages. Note that a shop can have any number of packages from 1 to 4**.
- The drop - location of each package is encrypted in a **QR code** placed at the **PN** of each shop **(in the CoppeliaSim scene)**.
- QR codes will NOT be present on the printed arena during the theme run.
- Note: For this theme, teams **DO NOT have to create any mechanical structure** for dropping the Packages as the robot does not actually drop any Package. It only needs to travel to the Delivery Junction and indicate the dropping of a Package. The method used to indicate this drop is discussed in the Theme Run section below.

E. Medicine Packages

- The packages stocked in the medical shops can be differentiated using their shape and colour.
- The packages can be any of the three shapes - **Square, Triangle, Circle.**
- The colour of the package can be any of the four colours - **Sky Blue, Pink, Orange, Green.**
- A shop deals with only SAME SHAPED packages i.e **a cell will not have multiple shapes**. All packages in a company will have UNIQUE COLORED packages i.e **no two packages in a cell will have the same colour.**

> *Along with the above mentioned parameters, you should also obtain the graph of the path (as in Task 3A) for implementing the path planning algorithm later while traversing the arena.*

3.2 CoppeliaSim Arena

3.2.1 Emulation

On the printed arena, the AlphaBot will be used for line following and package delivery. The AlphaBot's motions will be tracked by an emulation that runs in CoppeliaSim (in real-time). Since there is no mechanism on the robot for package pick-up and delivery, this should be indicated in the virtual arena. The CoppeliaSim scene should also include representations of other features such as traffic signals, roads under construction and packages. Refer sections 3.2.2 and 3.2.3 for more details.

3.2.2 Designing the CoppeliaSim Arena

You will receive a basic arena in CoppeliaSim that is identical to the printed arena's grid structure. For enhancing the virtual arena's aesthetics, ***Bonus points*** will be given. Refer to the Judging and Scoring section for more details.

By adding 3D models to the CoppeliaSim scene, you may create a cityscape on the virtual area. You will be given a collection of 3D models, which you can use for improving the aesthetics of the arena. You are free to design any other models if required. Make sure to incorporate **futuristic aspects** as imaginatively as you can, keeping in mind that the arena portrays a **smart city**.

You will be provided with:

- Arena with areas marked for adding 3D models
- 3D models of buildings, trees etc
- 3D model of a traffic signal
- 3D model for indicating roads under construction
- 3D models of packages (details are mentioned in section 3.2.3)

Your task is to:

- Place the given models of buildings (or any other model) in the arena for improving the aesthetics
- Design a storage mechanism on the robot for carrying the packages to the delivery locations
- Design any model for indicating the five medical shops in the arena

Ensure that the following points are satisfied:

- The grid structure of the arena should remain the same.
- You are **NOT allowed** to change the
  - Pattern and Width of the roads
  - Yellow patterns at nodes and their dimensions
  - Number of grids and naming of rows and columns
- The external design elements should not interfere with the movement of the AlphaBot model in the simulator.
- No models are placed outside the designated areas

> **You are Not Allowed to add any elements to the real arena during theme run.**

3.2.3 Setting Up the CoppeliaSim Arena

After extracting the arena parameters from the Test Image:

- The same scene configuration (as in Test Image) should be initially generated in the CoppeliaSim scene.The **traffic signals, start and end nodes** should be generated at correct nodes in the CoppeliaSim scene.
- All the roads under construction should be represented in the virtual arena.
- For the above mentioned parameters, 3D models will be provided and it is **mandatory** that they are indicated before the run starts.
- The **packages** should be generated in the respective shops at a position as given in Figure 4 (for the configuration given in Figure 2). The dimensions for keeping the packages is also shown in Figure 5.

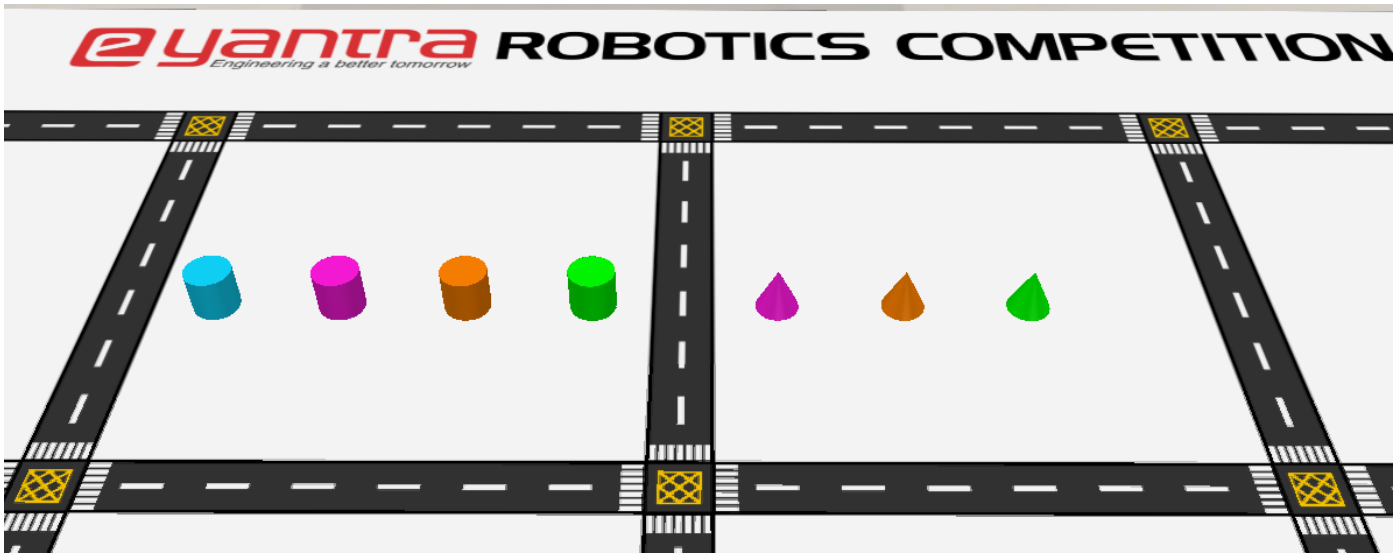

Figure 4: Placing packages in medical shop

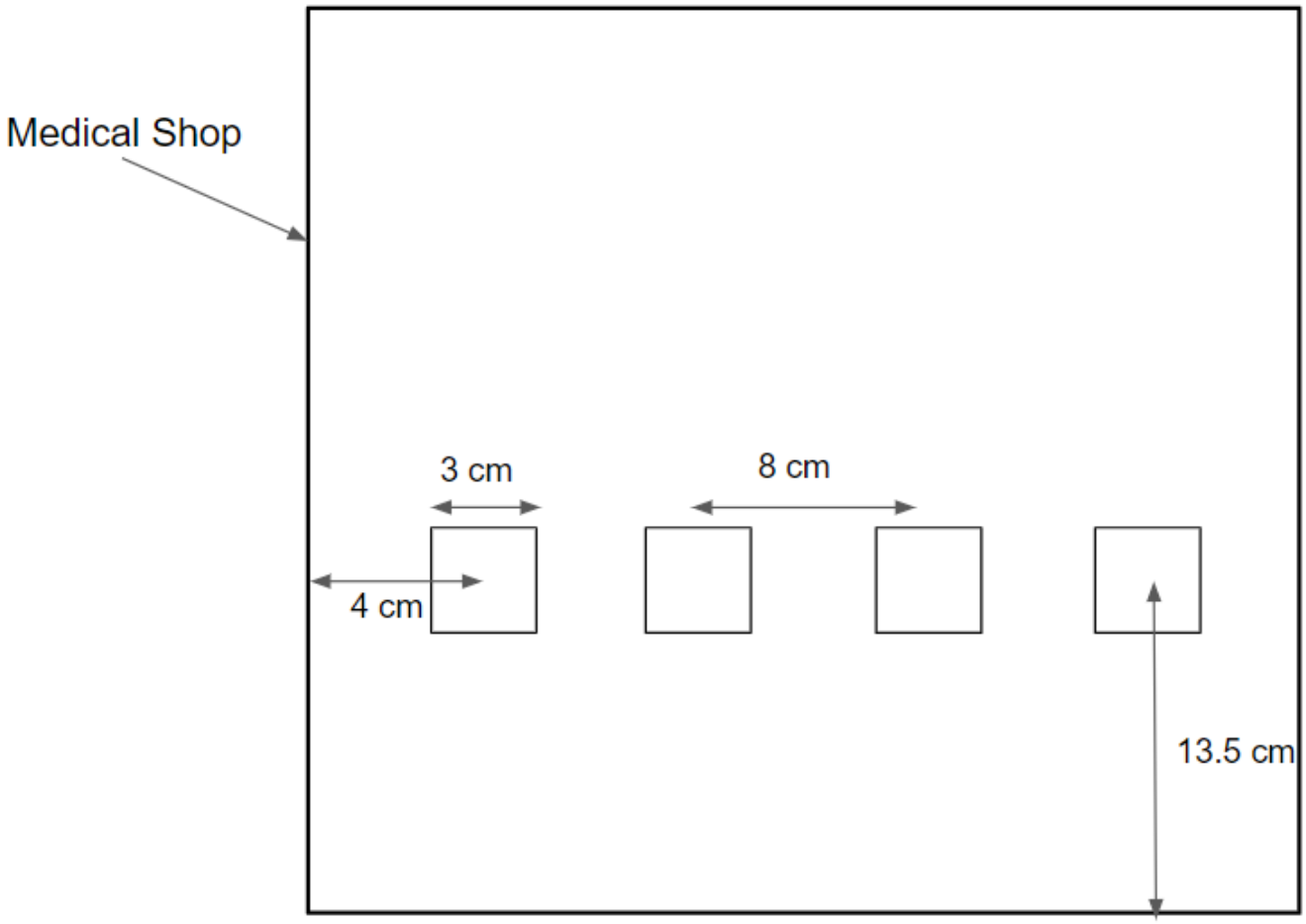

Figure 5: Position of packages

Ensure that you use only these designated areas so that the packages will not interfere with the movement of the robot near the medical shops. A library of 12 models (4 colours of each shape) will be provided and you **have to use the same models** while setting up the arena. The shape and its corresponding 3D structure can be found in Table 1.
The **colour of the packages must remain the same** as given in the test image.

Table 1 : Shapes and 3D models

Shape in Image	3D Model
Square	Cube
Triangle	Cone
Circle	Cylinder

- Five QR codes will be given along with the configuration image numbered according to the shop number it belongs to. If the QR code is named “1.png”, this will contain the delivery location for each package kept in Shop_1 as a dictionary. **You have to insert these QR codes to the PNs along with the other components** in the virtual arena.
- Once the above steps are completed, the CoppeliaSim arena is set up.

- Failing to represent any of the arena parameters will result in you losing marks. Refer to the Judging and Scoring section for more details.

3.3 Printed Arena

There are various steps involved in setting up the arena for a run.

A. Printing the arena

- The arena which is already printed (in Task 3) will be used for all theme runs.Image of the arena that was given for printing is shown in Figure 6.

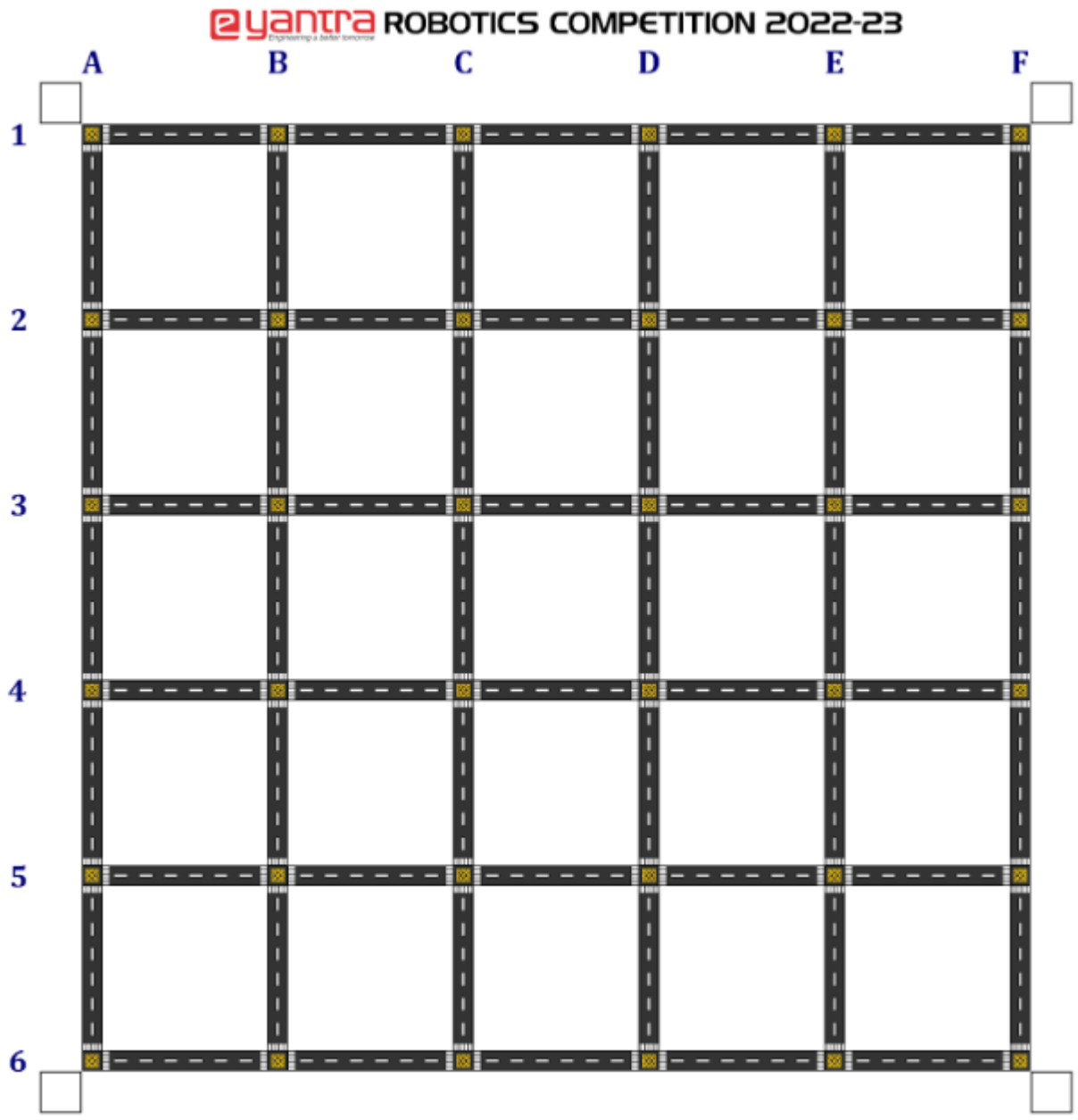

Figure 6: Arena Design

B. Setting up the overhead camera

Find a location first where there is **adequate lighting** and enough room to accommodate the arena properly. From this point on, you will be finishing the entire theme on this arena. Therefore, be careful to select the right area so that you won't have to shift the setup later. The camera (included in the kit) can be initially fixed at a height of **8 feet**, and the USB cable from the camera to the laptop can then be connected. Make sure you are receiving a proper feed from the linked camera by using the camera app on your laptop. Keep the printed arena in the camera's field of view now, and use your laptop to access the camera's feed. Before proceeding any further, ensure the following points are satisfied:

- The full arena is visible in the feed **without any obstructions** in between. No part of the arena should be cropped out in the obtained frame.
- If there is any part being cropped out, **adjust the height** of the camera so that the full arena is covered.
- All regions of the arena should receive the same amount of focus. The obtained arena image **should not have any blurring**.
- There is **sufficient lighting** where the arena is placed.

C. Placing the ArUco markers

- For the purpose of setting up the arena, the **ArUco images** (given in resources for Task 3C) should be printed and pasted on cardboard pieces. 1 ArUco must be pasted on 1 cardboard piece and hence you should have 5 ArUco images on 5 different cardboard pieces.
- Each cardboard piece should be then placed on four corners of the arena. The ID of the ArUco and its position on the arena should be as given in Figure 7.
- The fifth ArUco marker **(id = 5), should be mounted on top of the robot**. This marker will be used for locating the robot on the arena and for emulating the same in CoppeliaSim (as in Task 3C).

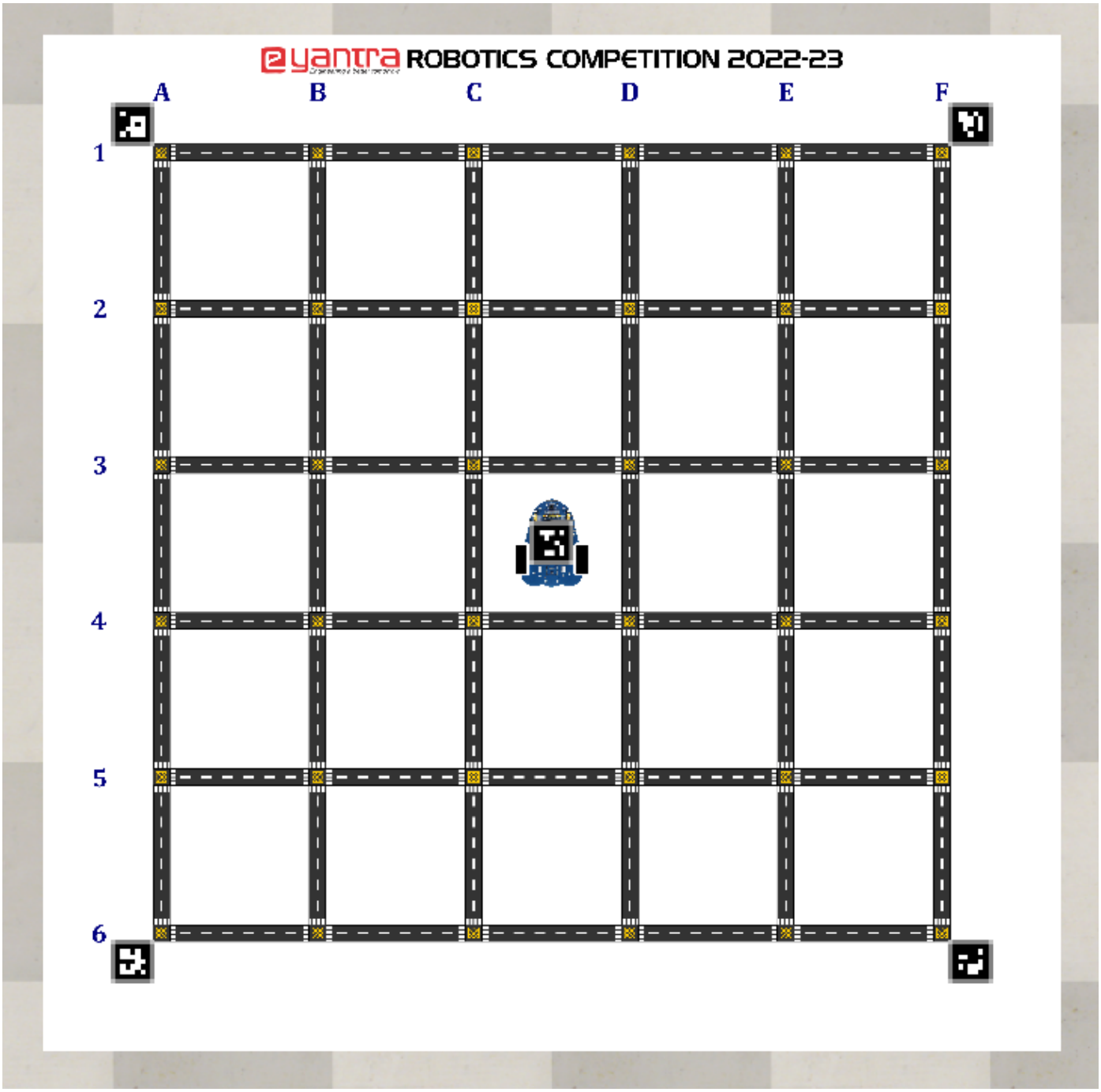

Figure 7: ArUco markers on Arena

4. Theme Run

- From the test image given, required information should be **extracted** and the CoppeliaSim arena should be **set up**.
- The robot will now start its **traversal from the start node** (in the real arena). Each movement of the robot should be correctly **emulated** in the CoppeliaSim scene for the full run.
- Once the robot reaches the first **PN** (*you may choose to visit the Shops in any order according to your path planning algorithm*) in the real arena, it will notify the CoppeliaSim scene. Now the **QR code will be scanned**

in the scene (by the vision sensor attached to the AlphaBot model) and **delivery location** for each package will be sent to the robot. This should be done using **socket programming**.

- Using the extracted information, the robot **plans its path** and starts traversing the arena.
- Since there is **no mechanism** for pick and place, we will be using **RGB LEDs** for indicating the pickup and drop of a package.
- For example, if the QR code has the delivery location for a green square, it means that then one of the LEDs should be made to emit Green light. The robot should stop and the colour of the package picked up should be indicated using the RGB LEDs. The colour, shape and delivery location of the picked up package should be printed on the console simultaneously in the following format: *"PICKED UP: colour, shape, delivery location"*. For example: **PICKED UP: Green, Square, D3**
- Note that as long as the Green LED is ON, it indicates that the package is still on the robot.
- Once it reaches the required delivery location, the LED **must** be turned OFF. This indicates that the package is no longer on the robot and it has been delivered. The colour, shape and delivery location of the delivered package should be printed on the console simultaneously in the following format: *"DELIVERED: colour, shape AT delivery location"*. For example: **DELIVERED: Green, Square AT D3**
- The package being delivered should be placed at the top left corner of the node in the direction which the robot is facing. If the robot's front is facing in the direction as shown in Figure 8, the package (whose delivery location is the node indicated here) should be kept at the location shown using the green square. You can refer to Task 2B output video for better understanding.

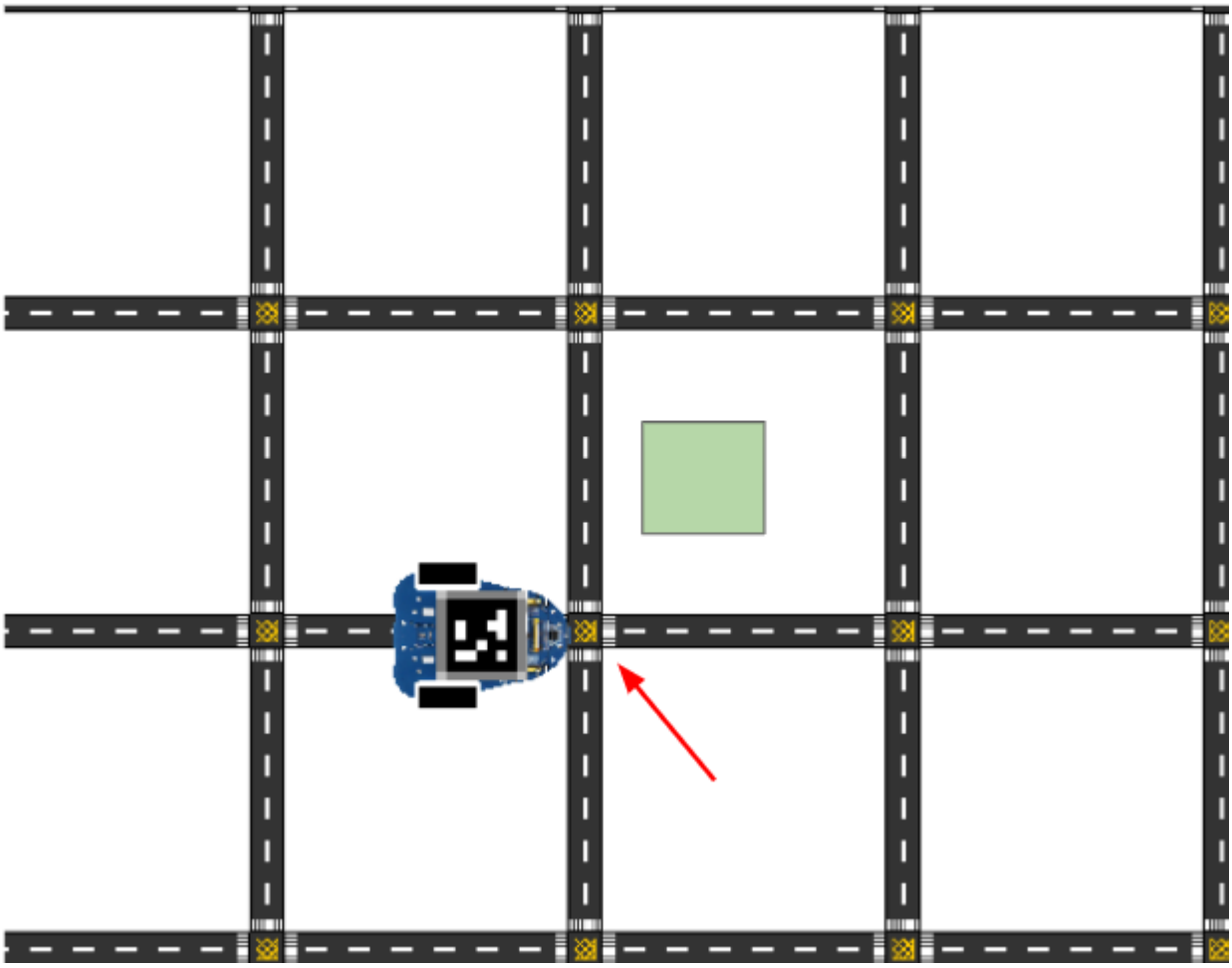

Figure 8: Package Delivery

- While delivering the package, you should make sure that the package placed **does not interfere** with other elements of the scene (like buildings or any other 3D model you might have used for designing the virtual arena).

---

5. Software Specifications

- e-Yantra has provided all teams with an e-Yantra's version of the Raspbian image. This image has all the softwares and drivers **pre-installed**.
- The teams must use **OpenCV and Python** to write their code. These software are already installed in the image file given to you (in Task 3B).
- 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. Hardware Specifications

A. Use of Raspberry Pi:

- All participating teams must use ONLY the Raspberry Pi B sent to them in the kit. Only one Raspberry Pi B is allowed per team.

B. Use of AlphaBot:

- All teams must use ONLY the Alpha Bot kit sent to them as the robot. **You do not need to purchase any other robotic kits or components for completing the theme**. However, they are allowed to create any type of mechanical mount for mounting the camera.
- Teams are **NOT** allowed to use any wireless remote or any other communication protocol or devices such as cameras, other than those provided in the kit.
- **There is NO need to build any mechanism (like robotic arm) for the pick and place of the packages.**

C. Components:

Along with the Raspberry Pi and Alpha Bot kit, teams have been given

- Web Camera and USB cable (for emulation)
- 18650 Batteries(4) and Battery Charger
- Jumper wires
- RBG LED(4)

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

---

7. 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 better score from the two runs will be considered as the team's score). A **maximum of three repositions** (explained below) will be allowed in each run.
- Test Image will be given before the run and after getting the required information from the image, the set up should be completed under the supervision of an e-Yantra official. The setup includes the **setting up of the virtual arena in CoppeliaSim and placing the robot at the start node**.
- A **shop** can have any number of packages **between 1 and 4.** The **maximum** number of packages that the **robot can carry** at a time is **3**.
- After the setup is completed, you can start the line following task (emulation should be started in parallel). The **timer** will be started.
- Once the run starts, **human intervention (other than for reposition) is NOT allowed**.
- After delivering all the required Packages, the robot should return to the End Node. A message **"End of Task"** should be printed on the console and the colour of all four **LEDs should be made White**.
- Robot is **NOT** allowed to traverse through the cell, it always has to **follow the black line** for traversal.
- A **run ends** and the timer is stopped when:

- The robot stops and end of the Task is indicated **OR**
- If the maximum time limit for completing the task is reached **OR**
- If the team needs repositioning but has used all 3 repositioning options of that run.

- Second run will start once again whilst resetting the score, timer and arena. The score of both runs will be recorded and the best of two runs will be considered as the team's score.

- Participants are not allowed to keep anything inside the arena other than the robot. The time measured by the reviewer will be final and will be used for scoring the teams.

- Time measured by any participant by any other means is not acceptable for scoring.

- 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.

7.1 Repositioning of robot

Reposition can happen in two situations:

While traversing the arena if the robot strays off the black line
OR
If the robot traverses through any road which is under construction

In any case, an e-Yantra official who is monitoring the task will place the robot on the previous node (node previously traversed by the robot prior to straying off the black line/ node before which the robot went to under construction road).
This is termed as a **Reposition**.

- Note that the timer used for measuring the task completion time in the competition will be continuously running during a Reposition and the robot will not be switched off.
- A **maximum of three Repositions** will be allowed in each run.
- Note that each **reposition will cause a penalty**. Details can be found in the Judging and Scoring section.

---

8. Judging and Scoring

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

> **Score = ((600 - T) * 0.5) + (CP * 100) + (CD * 100) - (TP * 50) - (RP * 50) + SS + ES + B + DB**

Valid Run: Given a theme configuration, a run is considered valid only when the robot starts from the start node, **picks up and delivers at least 2 required packages correctly** to their target locations and stops at the end node within 600 seconds.

Table 2 : Formula Parameters

Parameter	Description
T : Time taken	This is the time taken for the run completion in seconds. A run will be considered complete in any one of the following conditions: i) All packages are delivered and the robot stops at the end node (in 10 minutes) ii) Run exceeds 600 seconds (10 minutes) iii) A fourth reposition is required
CP : Correct Pickup	CP count is incremented only if all of the the following criteria are met while picking up a package: i) Glowing correct coloured LED ii) Printing colour, shape and delivery location of the picked up package to console in the format: "PICKED UP: colour, shape, delivery location" iii) Showing pick up in CoppeliaSim Note: Failing to satisfy any one of the three criteria will lead to CP=0 for that particular package
CD : Correct Delivery	CD count is incremented only if all of the following criteria are met while delivering a package: i) Package is delivered to correct location ii) Switching off correct coloured LED iii) Printing to console: "DELIVERED: colour, shape AT delivery location" iv) Showing delivery in CoppeliaSim Note: Failing to satisfy any one of the four criteria will lead to CD=0 for that particular package
TP : Traffic signal Penalty	A penalty TP is incurred each time the robot misses a traffic signal. A robot misses a traffic signal when it does not stop at the traffic signal node for 5 seconds
RP : Reposition Penalty	A penalty RP is incurred each time a reposition happens. This can be reposition due to either of the situations mentioned in section 7.1. A maximum of 3 repositions are allowed (with penalty). A fourth reposition will lead to end of the run
SS : Setup Score	A Setup Score (on 50) will be awarded for correctly setting up the virtual arena based on the given test image. Full points will be awarded if i) All packages are placed in the correct location. The colour of packages should remain the same. ii) All traffic signals and roads under construction are properly indicated. iii) Packages are delivered without any collision with other scene objects in CoppeliaSim.
ES : Emulation Score	An Emulation Score (on 50) will be awarded depending on how well the emulated robot follows the actual robot. This parameter will be **evaluated subjectively**. The score given will be proportionate to the accuracy of emulation The score will be higher if the position and orientation of the robot are emulated with higher accuracy. Marks will be reduced if collision happens between the robot and any other scene elements.

Parameter	Description
B : Bonus	A Bonus (B) of 200 points awarded in a run when: i) No Penalty (TP or RP) is incurred ii) All required packages are correctly picked up and delivered Failing to satisfy any one of the above two conditions will result in zero bonus points
DB : Design Bonus	A Design Bonus (on 100) will be awarded for the beautification of the arena such as placing buildings, shops etc abiding to the conditions mentioned in the Designing the CoppeliaSim arena section. This is a **purely subjective** score and the score given will depend on how the aesthetics of the scene is improved.

1 Like

[Announcement] PB RuleBook Released

[Important] Hints and tips for theme implementation (Task 5 and 6)