e-Yantra Robotics Competition - 2020-21
Nirikshak Bot
Task 4C - Theme and Implementation Analysis

NB_534

Team Leader Name	Aakash Mangla
College	Bharati Vidyapeeth College of Engineering
Team Leader Email	Aakashmangla15@gmail.com
Date	7 Jan 2021

Scope

Q1. State the scope of the theme assigned to you. (5)

< Teams should briefly explain in their own words the theme assigned. What in your opinion is the purpose of such an application? You may use figures/diagrams to support your answer (Make neat and labelled diagrams).
Answer format: Text - limit: 100 words. >

Ans. Our theme Nirikshak Bot is based on quality assurance. Whenever a product is manufactured, the quality of the product is very important to prevent faulty products and costumer complaints after the delivery of product. Instead of humans, this quality assurance is done by robots. The product undergoes various test cases and challenges and if it passes the tests then the product is approved. Similarly we are testing the ball balancing platforms that are manufactured by making them go through tests which include passing a ball through maze on the top of the ball balancing platform in minimal time and least/no collisions. Overall this reduces errors and defects in products leading to high quality produce.

Testing your knowledge (Theme and Rulebook analysis)

Q2. Consider the following dictionary written in ball_details.json file:

```
{
   "red"      : ["T3_CB1"],
   "green"    : ["T2_CB2", "T1_CB1"],
   "blue"     : ["T1_CB3", "T3_CB3"]
}
```

Based on the dictionary given above, write the correct Collection Box for the following sequence of balls dispensed by BD: (5)

< This question is to check if you have understood how to interpret the ball_details.json file correctly. Hence fill in the answers carefully in the table below>

Sequence	Color	Collection Box Name
4th	Green	**T1_CB1**
5th	Blue	**T3_CB3**
2nd	Blue	**T1_CB3**
3rd	Red	**T3_CB1**
1st	Green	**T2_CB2**

Q3. Consider the JSON configuration given in Q2.

a) **What are the ENTRY and EXIT cell coordinates used by the *first green ball* for all the tables it is passing through? (2)**

b) **What are the ENTRY and EXIT cell coordinates used by the *second blue ball* for all the tables it is passing through? (2)**

c) **What are the ENTRY and EXIT cell coordinates used by the *first red ball* for all the tables it is passing through? (2)**

< This question is to check if you have understood Arena section of the Rulebook. Write your answers point wise for (a), (b) and (c) >

Ans.

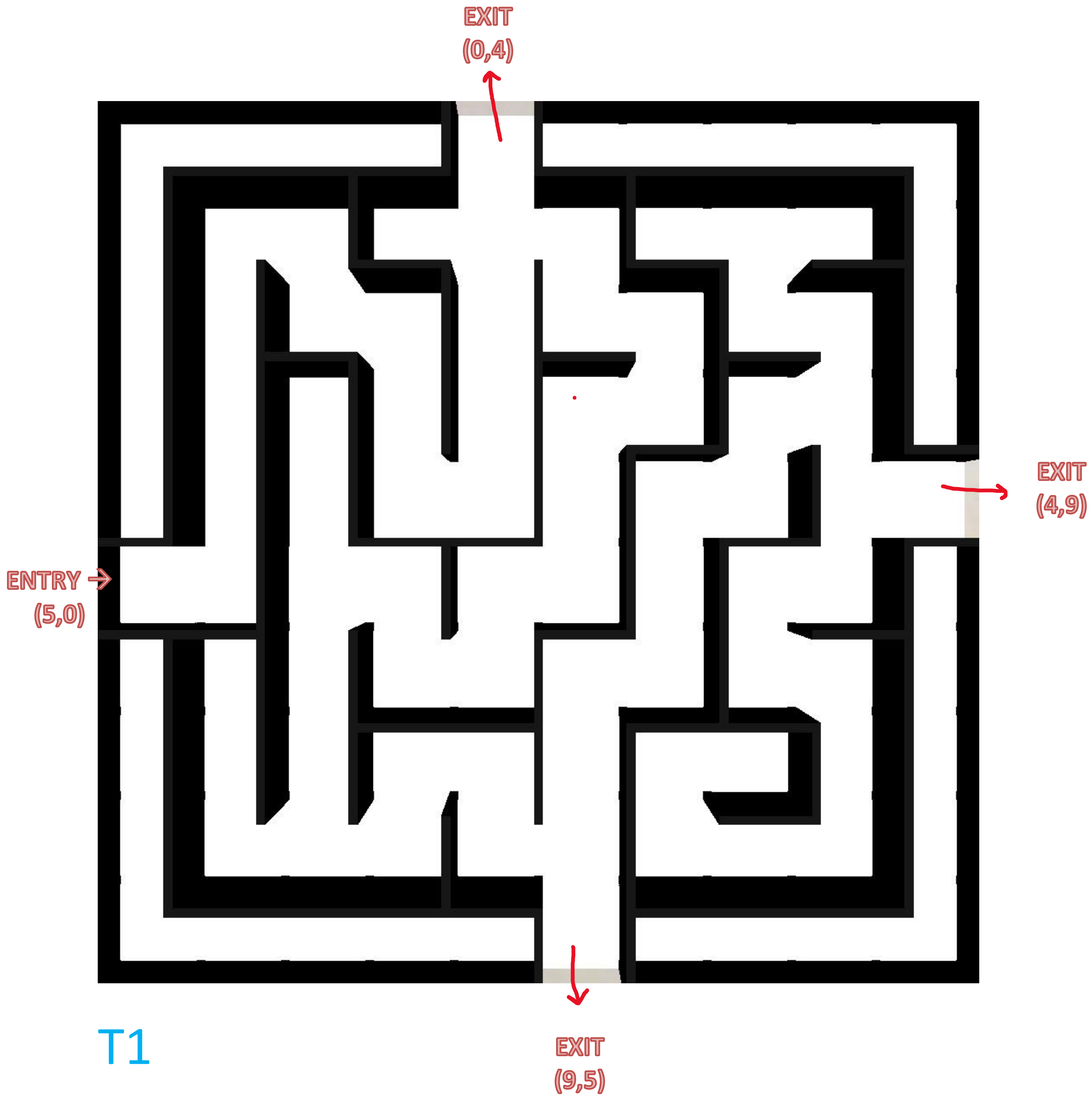
(a) T4: [ENTRY]-(0,5) [EXIT]-(9,4), T2: [ENTRY]-(0,4) [EXIT]-(9,4)

(b) T4: [ENTRY]- (0,5) [EXIT]-(4,0), T3: [ENTRY]-(4,9) [EXIT]-(0,4)

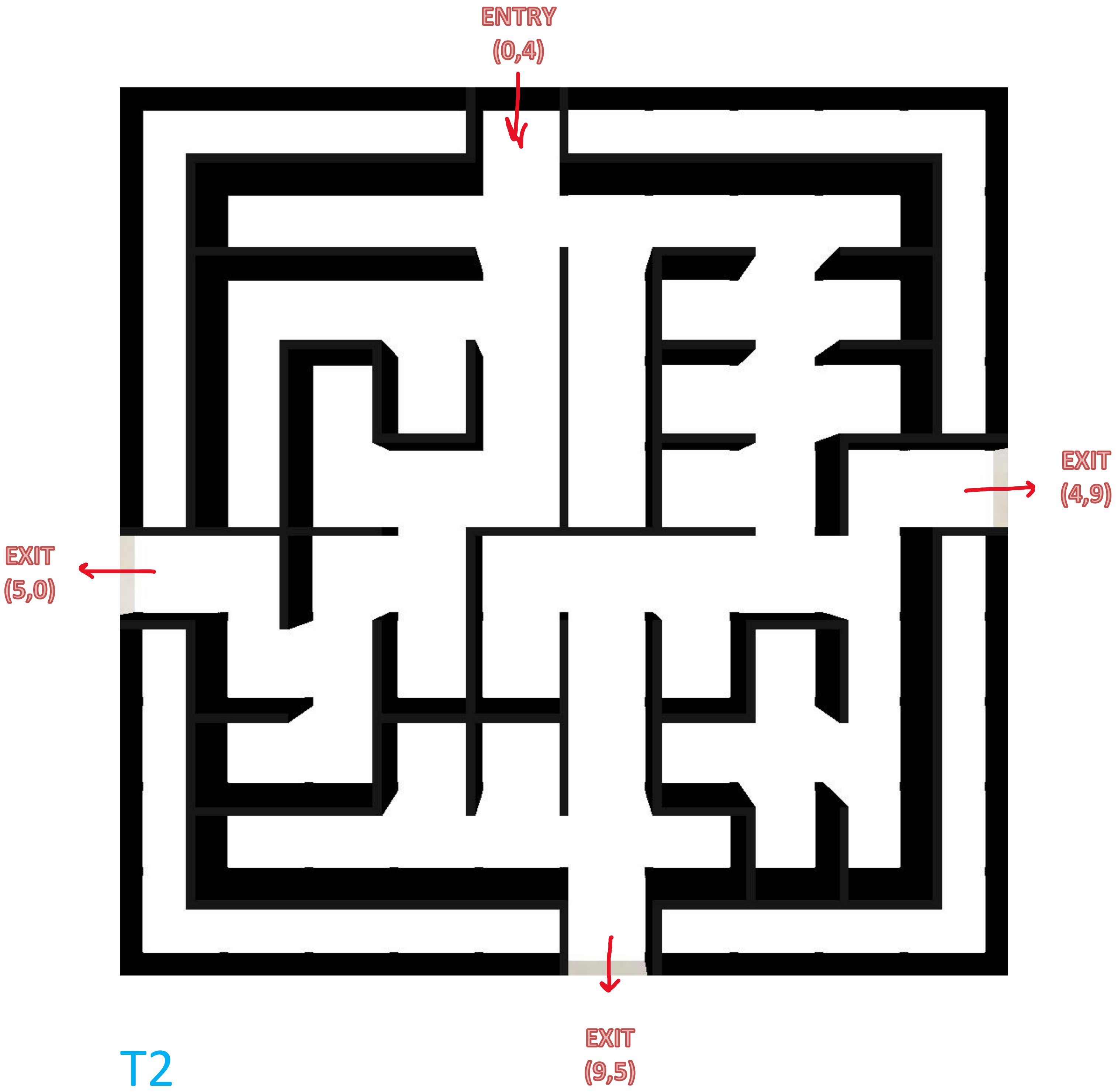
(c) T4: [ENTRY]- (0,5) [EXIT]-(4,0), T3: [ENTRY]-(4,9) [EXIT]-(9,5)

Q4. Download the *task_4c_maze_images.zip* file from this link (from Task 4C page). The images have been named maze_t1.jpg, maze_t2.jpg and so on (according to the Theme Run Requirements part under Theme Description section of the Rulebook). Generate these mazes on the single Platform Table one by one according to the resultant maze images shown in Figure 10 and 12 of Arena section in CoppeliaSim and capture a top-view screenshot for all of them. (4)

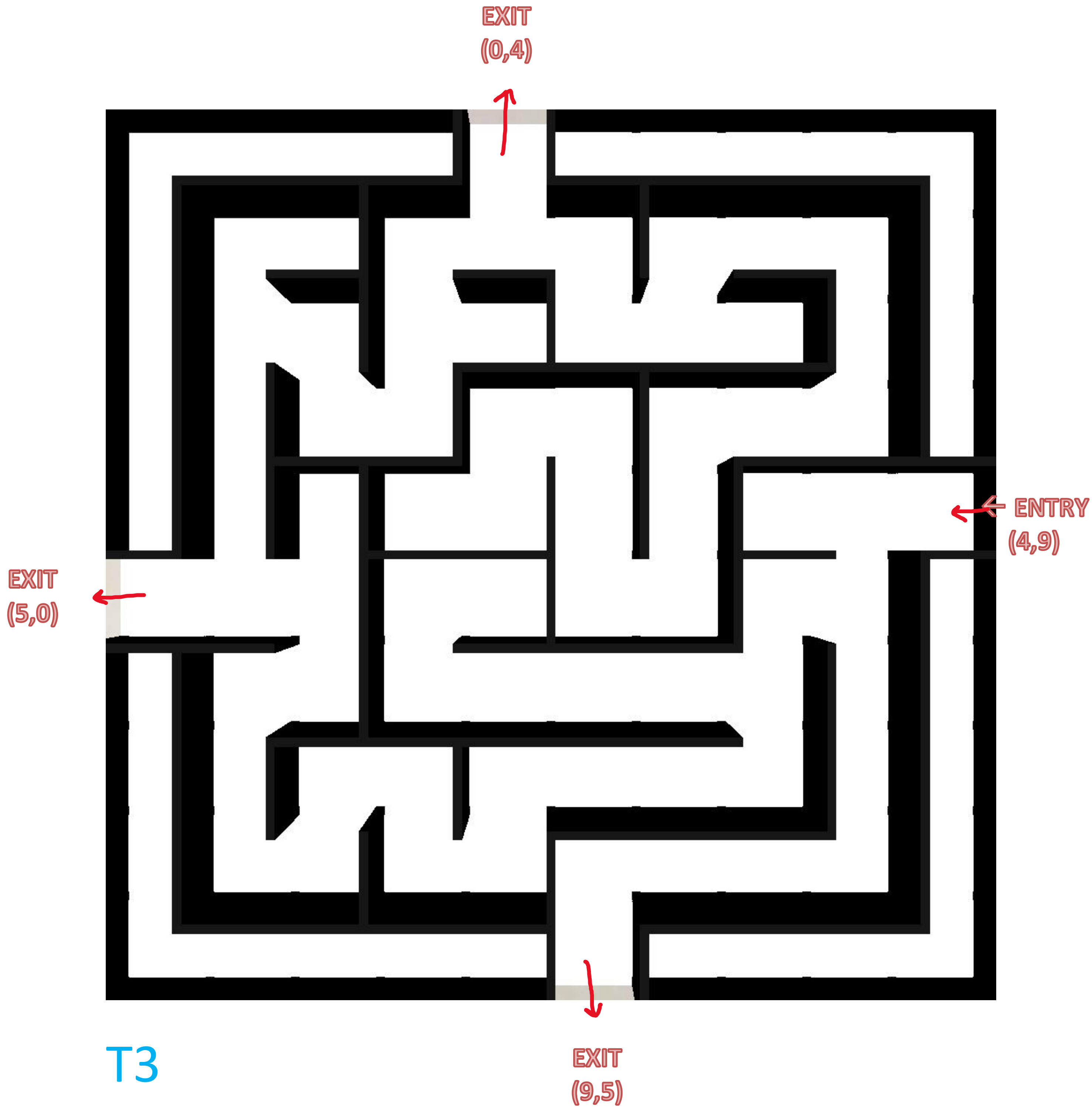
< Make sure to carve the respective EXIT points for all the mazes on Platform Table. Paste all the screenshots in this document. All the screenshot images should be properly labelled with ENTRY and EXIT clearly marked>



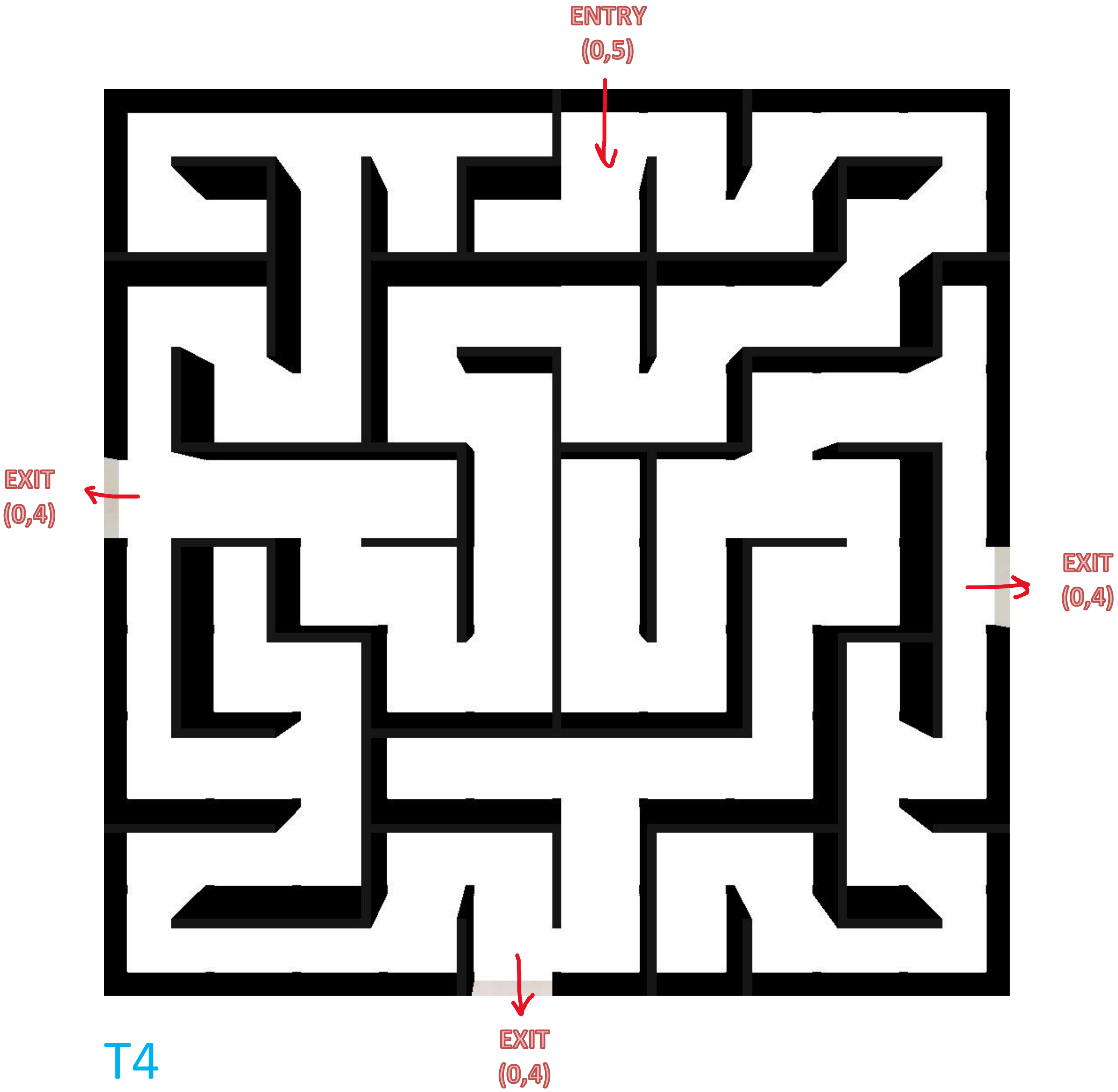
T1



T2



T3



T4

Q5. Consider the following table showing the scenario for each ball and calculate the final score: (5)

	CI	CP	CD	CT_4	CT_x	TB_4	TB_x	HP
1st	0	1	1	1	1	1	1	1
2nd	1	1	1	1	1	5	10	8
3rd	0	1	0	1	1	3	13	4
4th	1	0	0	1	1	10	20	15
5th	1	1	0	1	1	2	17	2

CM1	CM2	CM3	CM4
1	1	0	1

< Show your calculations in detail below>

Ans.

⇨ (0*10) + (1*100) + (1*50) + (1*1*100) + (1*1*1*100) + (0*1*1*1*10) + (0*1*1*1*10) – (1*10) = 340

⇨ (1*10) + (1*100) + (1*50) + (1*1*100) + (1*1*1*100) + (1*1*1*5*10) + (1*1*1*10*10) – (8*10) = 430

⇨ (0*10) + (1*100) + (0*50) + (1*1*100) + (1*0*1*100) + (0*1*1*3*10) + (0*0*0*13*10) – (4*10) = 160

⇨ (1*10) + (0*100) + (0*50) + (1*1*100) + (1*1*1*100) + (1*0*1*10*10) + (1*0*1*20*10) – (15*10) = 60

⇨ (1*10) + (1*100) + (0*50) + (1*1*100) + (1*0*1*100) + (1*1*1*2*10) + (1*0*0*17*10) – (2*10) = 210

⇨ (340+430+160+60+210) + (3*50) = 1350

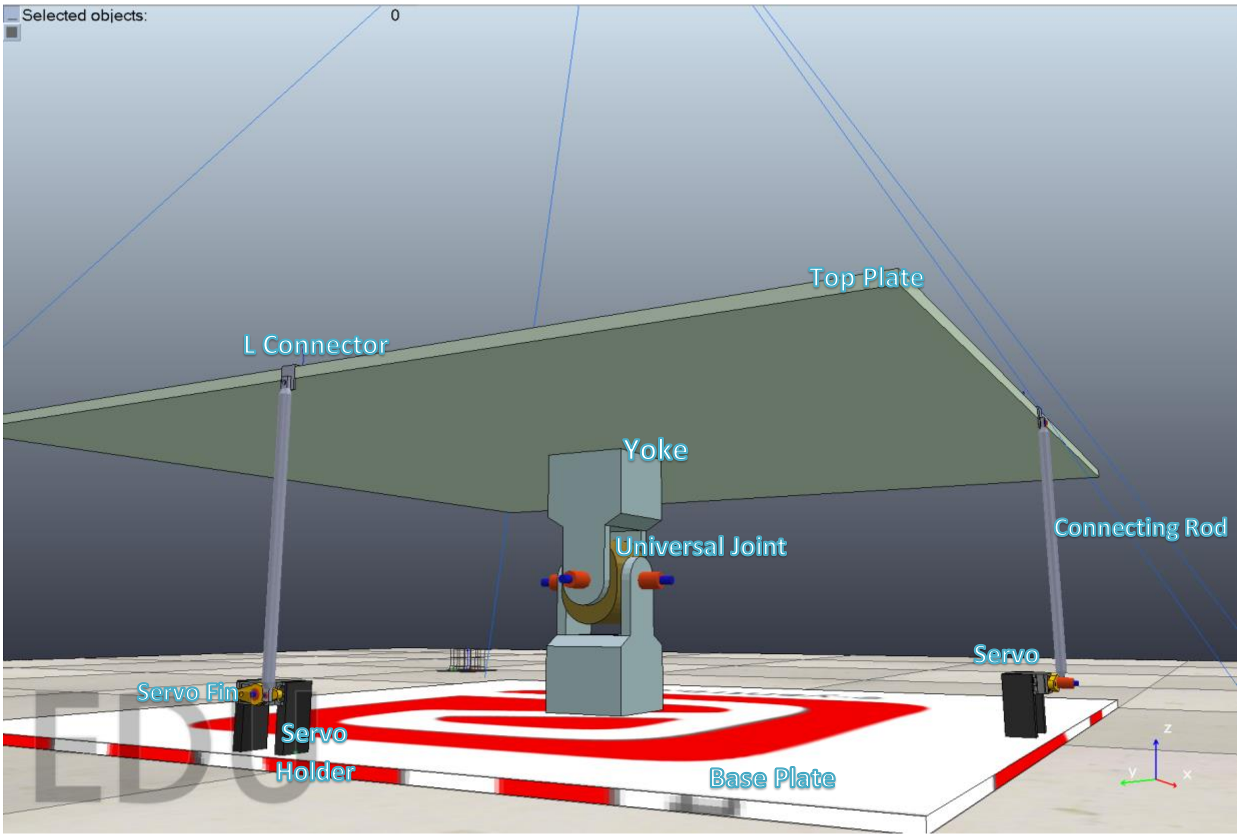
Mechanism

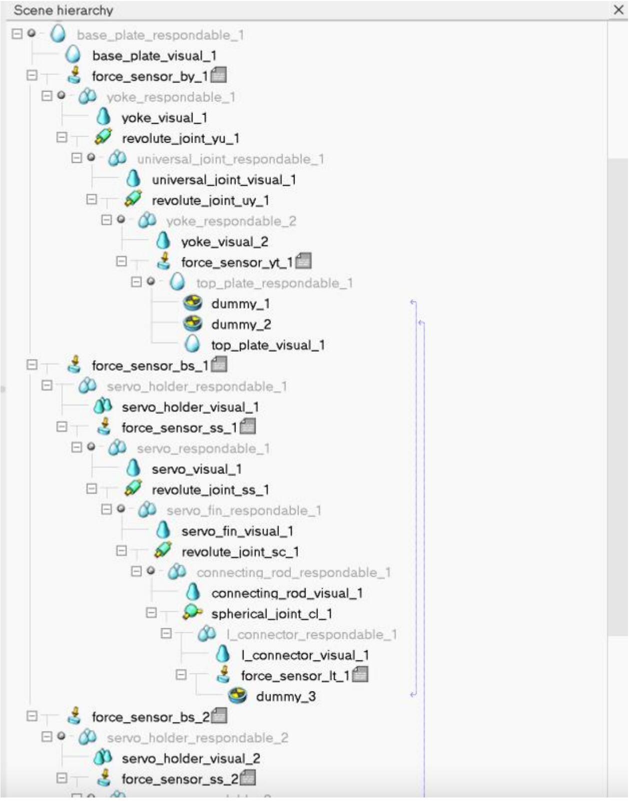
Q6. Explain the mechanism that you used for your ball balancing platform. (5)

< You must explain the mechanical construction of your ball balancing platform and how have you connected all the different components provided to you. Make properly labelled

diagrams to show the same. You may also use screenshots of the CoppeliaSim scene to demonstrate your mechanism.>

Ans.





```
base_plate_respondable_1
  base_plate_visual_1
  force_sensor_by_1
    yoke_respondable_1
      yoke_visual_1
      revolute_joint_yu_1
        universal_joint_respondable_1
          universal_joint_visual_1
          revolute_joint_uy_1
            yoke_respondable_2
              yoke_visual_2
              force_sensor_yt_1
                top_plate_respondable_1
                  dummy_1
                  dummy_2
                  top_plate_visual_1
  force_sensor_bs_1
    servo_holder_respondable_1
      servo_holder_visual_1
      force_sensor_ss_1
        servo_respondable_1
          servo_visual_1
          revolute_joint_ss_1
            servo_fin_respondable_1
              servo_fin_visual_1
              revolute_joint_sc_1
                connecting_rod_respondable_1
                  connecting_rod_visual_1
                  spherical_joint_cl_1
                    l_connector_respondable_1
                      l_connector_visual_1
                      force_sensor_lt_1
                        dummy_3
  force_sensor_bs_2
    servo_holder_respondable_2
      servo_holder_visual_2
      force_sensor_ss_2
```

We were provided with various components and using those components we were able to create the ball balancing platform. We first grouped 2 servo holders and connected them to base plate using force sensor, then using another force sensor we connected the servo to its holder. Then we put the servo fin and connected it to servo using a revolute joint as we need the servo fin to rotate to move the top plate. Then we connected the connecting rod to servo fin using another revolute joint and then finally connecting the rod to Top Plate using L connector and a force sensor. L connector was connected to rod using a spherical joint.
2 of such combinations of these components are used in our design as it can been seen in the image. We are using one each for both axes (X and Y). They are placed at the centre of the edge of the plate for x and y axis. Also a combination of 2 yokes and a universal joint is used to keep the design stable due to the heavy weight of the top plate. They are connected to each other using revolute joints and connected to top and base plate using force sensors. So using this mechanism we were able to balance a ball on the top plate as when the motor is enabled on the revolute joint used on servo and servo fin. We can rotate the fin to move the plate along X and Y axes (since we are using 2 servos).

Q7. In Task 1C, you were given the task to design the ball balancing platform while in Task 3, you were given the task to use this ball balancing platform to control the position of the ball on top of it. How did your ball balancing platform change between these tasks? (5)

< Explain in brief how your design changed in the subsequent tasks. If your design did not change, then justify your reasons for the same.
Answer format: Text - limit: 100 words. .>

Ans. Our platform changed drastically between these tasks. In our previous model we used 2 sets of universal joint and yoke pairs and 4 sets of servos. Whereas we simplified our design in Task3 by using only 1 set of universal joint and yoke pair and 2 sets of servos. We also changed the position of universal joint and yoke pair to the centre of the design. Also in previous design we placed the servos at the corners of base plate but in new design we moved them to the centre of the edge of the base plate. These changes were made to simplify our design.

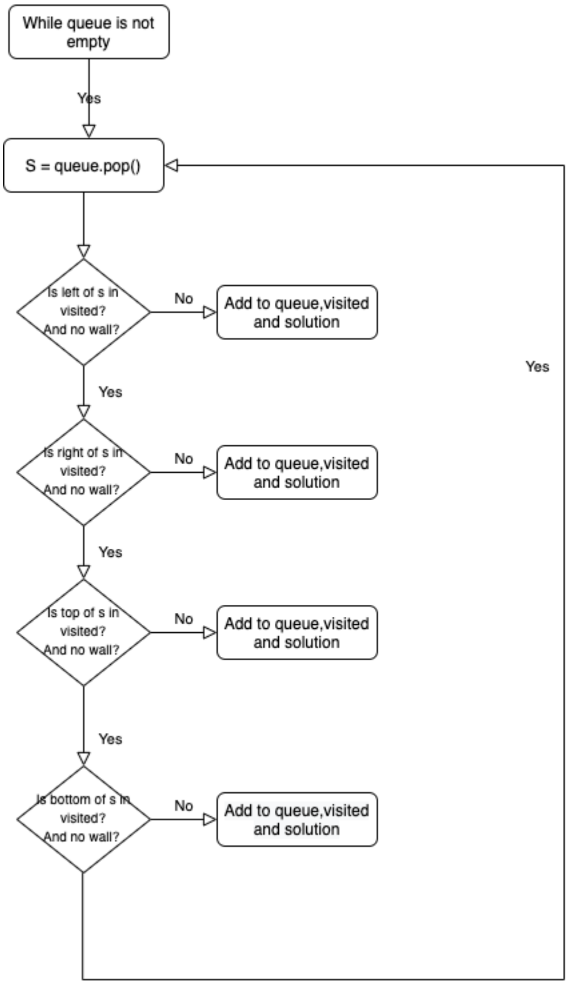
Path Planning

Q8. What kind of path planning algorithm did you use for finding the shortest path for the given maze images (in Task 4A)? (5)

< Explain the logic behind the algorithm and the reason for your choice if any. You can use a pseudo-code and/or flowcharts to help elucidate your answer. >

Ans. We used **breadth first search (BFS) Algorithm** for finding the shortest path for the given mazes. In this we basically visit all the possible cells in the maze from start till we reach the desired end cell. Then we traverse the back to the starting point from end point giving us the shortest possible path.

Flow Chart:


While queue is not empty
Yes
S = queue.pop()
Is left of s in visited? And no wall?
No
Add to queue,visited and solution
Yes
Is right of s in visited? And no wall?
No
Add to queue,visited and solution
Yes
Is top of s in visited? And no wall?
No
Add to queue,visited and solution
Yes
Is bottom of s in visited? And no wall?
No
Add to queue,visited and solution
Yes

This loop will run till the queue is empty and we reach the final end coordinate, after that using the solution dictionary which contains key value pairs for parent and child, we will start **backtracking** the route from end coordinate. And then after reversing that list we will get the shortest path.

Backtracking:

```
while (x,y) != (start_coord[0], start_coord[1]):
    try:
        path.append(solution[x,y])
        x,y = solution[x, y]
    except:
        path.pop(0)
        path = None
        break
```

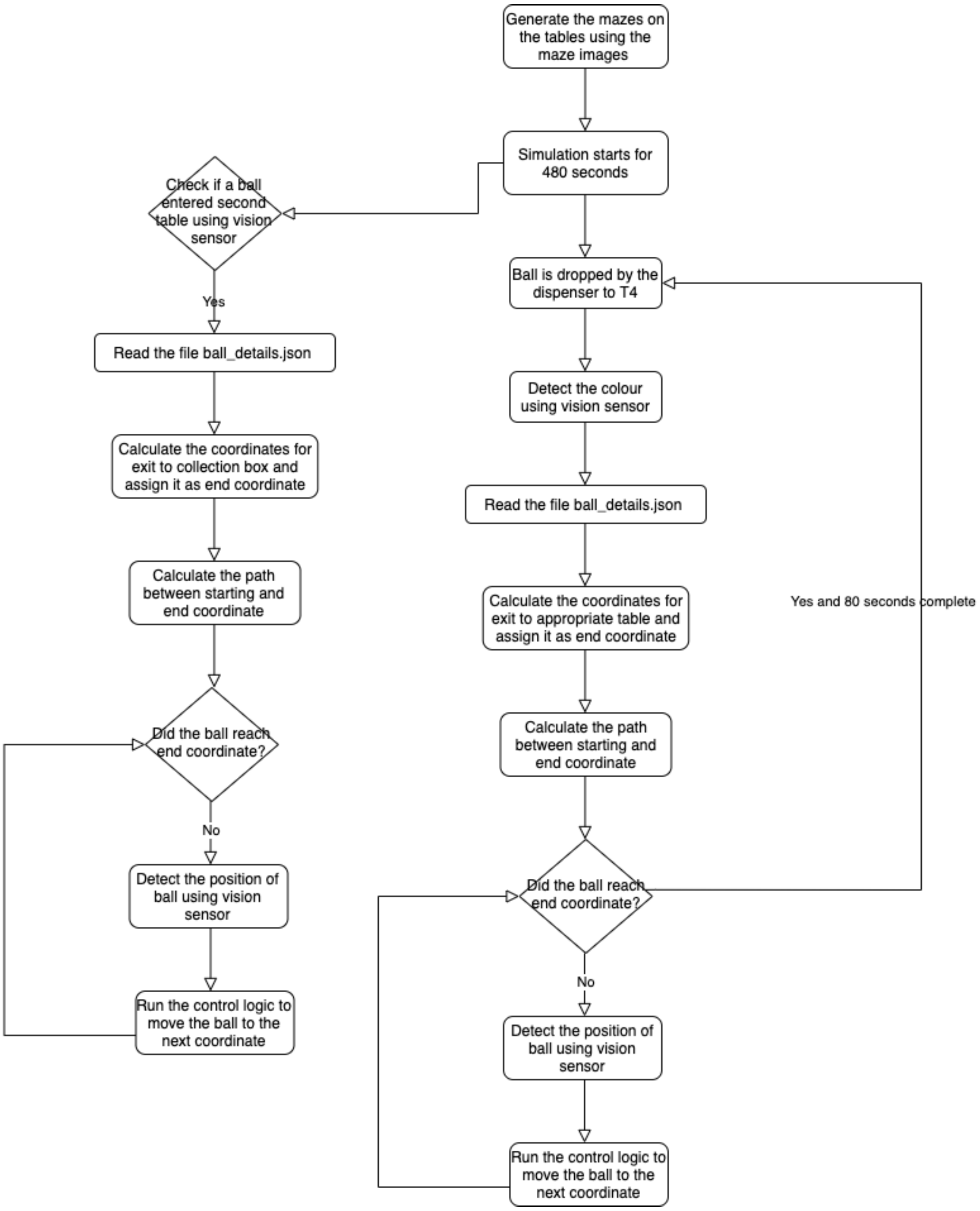
Algorithm Analysis

Q9. Draw a flowchart illustrating the algorithm / strategy you propose to use for theme implementation. (7)

< The flowchart should elaborate on every possible function that you will be using for completing all the Theme Run.

Follow the standard pictorial representation used to draw the flowchart. >

Ans.


Generate the mazes on the tables using the maze images
Simulation starts for 480 seconds
Check if a ball entered second table using vision sensor
Yes
Read the file ball_details.json
Calculate the coordinates for exit to collection box and assign it as end coordinate
Calculate the path between starting and end coordinate
Did the ball reach end coordinate?
No
Detect the position of ball using vision sensor
Run the control logic to move the ball to the next coordinate
Ball is dropped by the dispenser to T4
Detect the colour using vision sensor
Read the file ball_details.json
Calculate the coordinates for exit to appropriate table and assign it as end coordinate
Calculate the path between starting and end coordinate
Yes and 80 seconds complete
Did the ball reach end coordinate?
No
Detect the position of ball using vision sensor
Run the control logic to move the ball to the next coordinate

This flowchart explains the strategy we are planning to implement the theme.

Challenges

Q10. What are the major challenges that you have faced till now and the ones that you can anticipate in addressing this theme and how do you propose to tackle them? (3)

< Answer format: Bullet points

1. Challenge 1
2. Challenge 2
3. Challenge 3, etc. >

Ans. 1. Designing the control logic was a major challenge faced by our team but with the help of the resources provided by eyantra team we were able to design a decent algorithm for it.
2. Detecting the maze and converting it into an array was also a challenge but we did it successfully by spending enough time on it.
3. The major challenge we anticipate is completing the traversal of a ball on 1 maze in less than 80 seconds for any given exit. We will tackle it by improving our control logic.

IMPORTANT:

- **The document you submit should be in YOUR OWN WORDS. To avoid any copyright violations, you must NOT copy phrases directly from manuals or the web.**
- **The team should NOT mail or upload the document anywhere else except on the portal.**
- **Teams failing to submit the document by the deadline will lose the marks for this task.**
- **e-Yantra WILL NOT entertain any request for extension of deadline for uploading the task.**
- **e-Yantra holds complete discretion to disqualify a team if any foul play is suspected.**