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

<NB_2650>

Team Leader Name	Himanshu Thakur
College	National Institute Of Technology , Kurukshetra
Team Leader Email	himanshuthakur8119@gmail.com
Date	14/1/2021

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. >

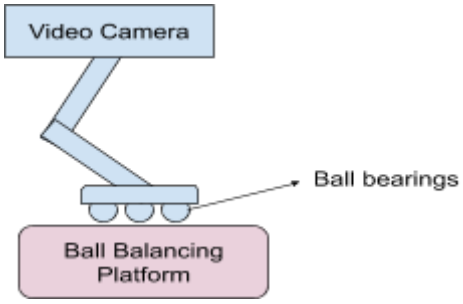
The theme involves delivery of balls through a conveyor belt, to a maze placed on the ball balancing platform. The ball balancing platform has to ensure that the ball traverses the correct path and exits the maze in minimum time and with minimum collisions with the maze walls. The quality depends mainly on:

1. Sensitivity of the balancing platform,i.e. how well it responds to minute changes in ball position(ensured through platform design)
2. Precision of PID control; a fine tuned system will produce least errors and collisions.(ensured through logic for ball traversal and PID constants)

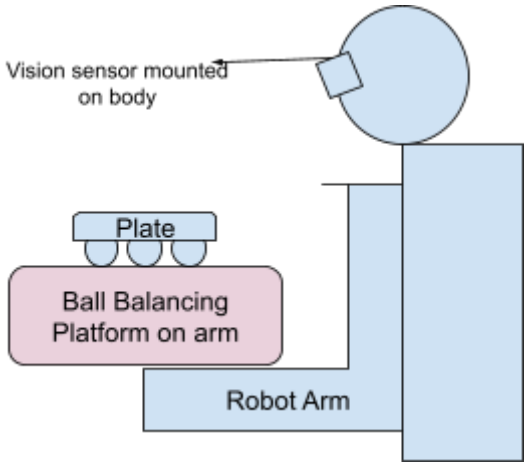
Hence a system with better value of the above two factors will perform better in the QA test.

Applications:

3. Stabilising video recording in a moving vehicle by attaching a camera (with ball bearings on base) on the platform.



4. Personal robot or waiter robot which places items on a plate ,placed on the platform, to prevent spilling.



5. Visually appealing and entertaining way to demonstrate Optimal Control.
6. Ping pong robot, which shoots the ball upwards after balancing and then balances it again when it falls back.

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	
5th	Blue	
2nd	Blue	
3rd	Red	
1st	Green	

Answer:

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)>

Answer:

(a) First Green Ball : It passes through table 4 and table 2

Table 4 => Entry Coordinate: (0,5)
Exit Coordinate: (9,4)

Table 2 => Entry Coordinate: (0,4)
Exit Coordinate: (9,5)

(b) Second Blue Ball : It passes through table 4 and table 3

Table 4 => Entry Coordinate: (0,5)
Exit Coordinate: (4,0)

Table 3 => Entry Coordinate: (4,9)
Exit Coordinate: (0,4)

(b) First Red Ball : It passes through table 4 and table 3

Table 4 => Entry Coordinate: (0,5)
Exit Coordinate: (4,0)

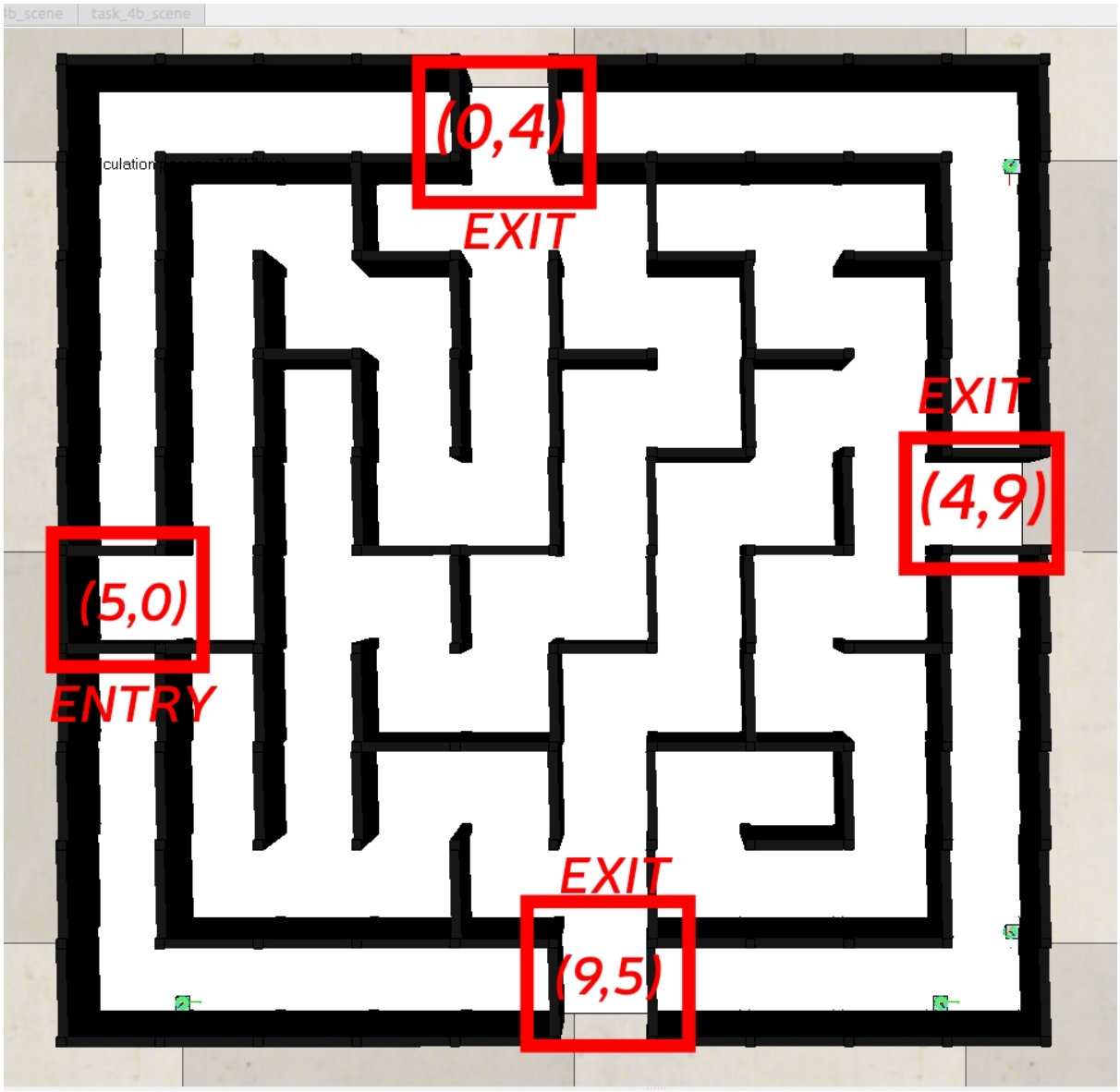
Table 3 => Entry Coordinate: (4,9)
Exit Coordinate: (9,5)

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)

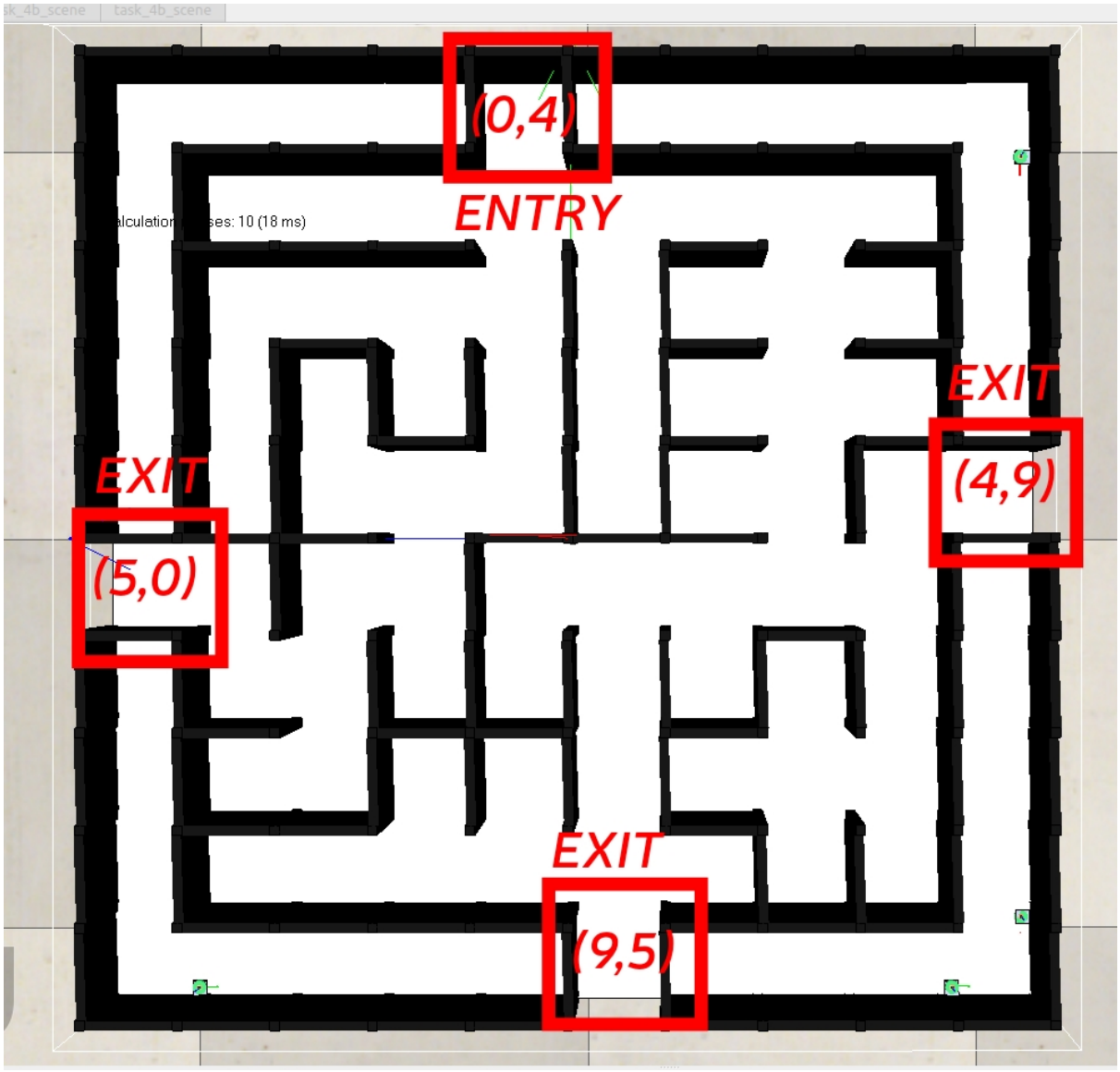
< Make sure to carve the respective EXIT points for all the mazes on Platform Table. Paste all the screenshots in this document. All the screenshot images should be properly labelled with ENTRY and EXIT clearly marked>

Answer:

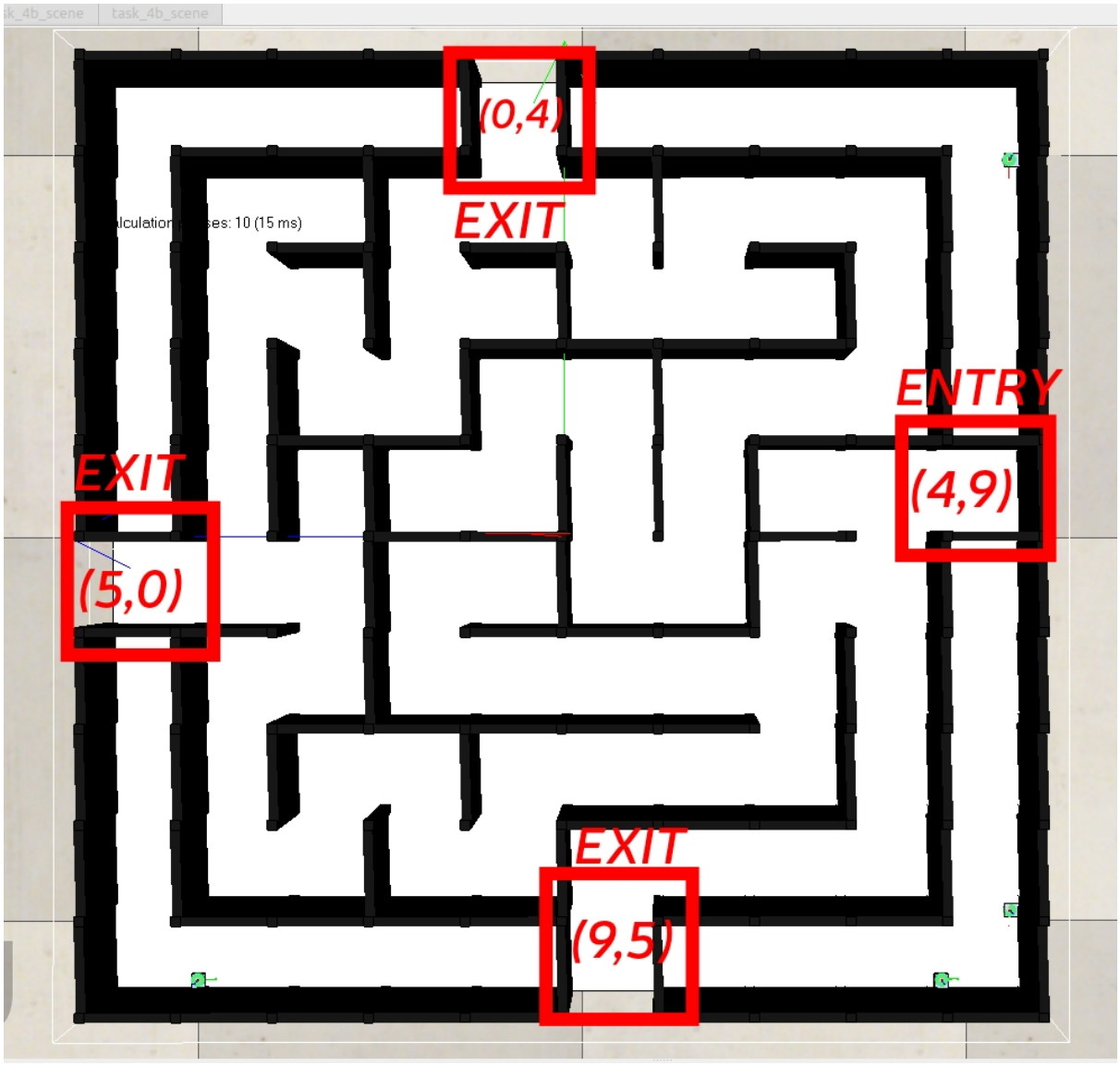
(I) maze_t1



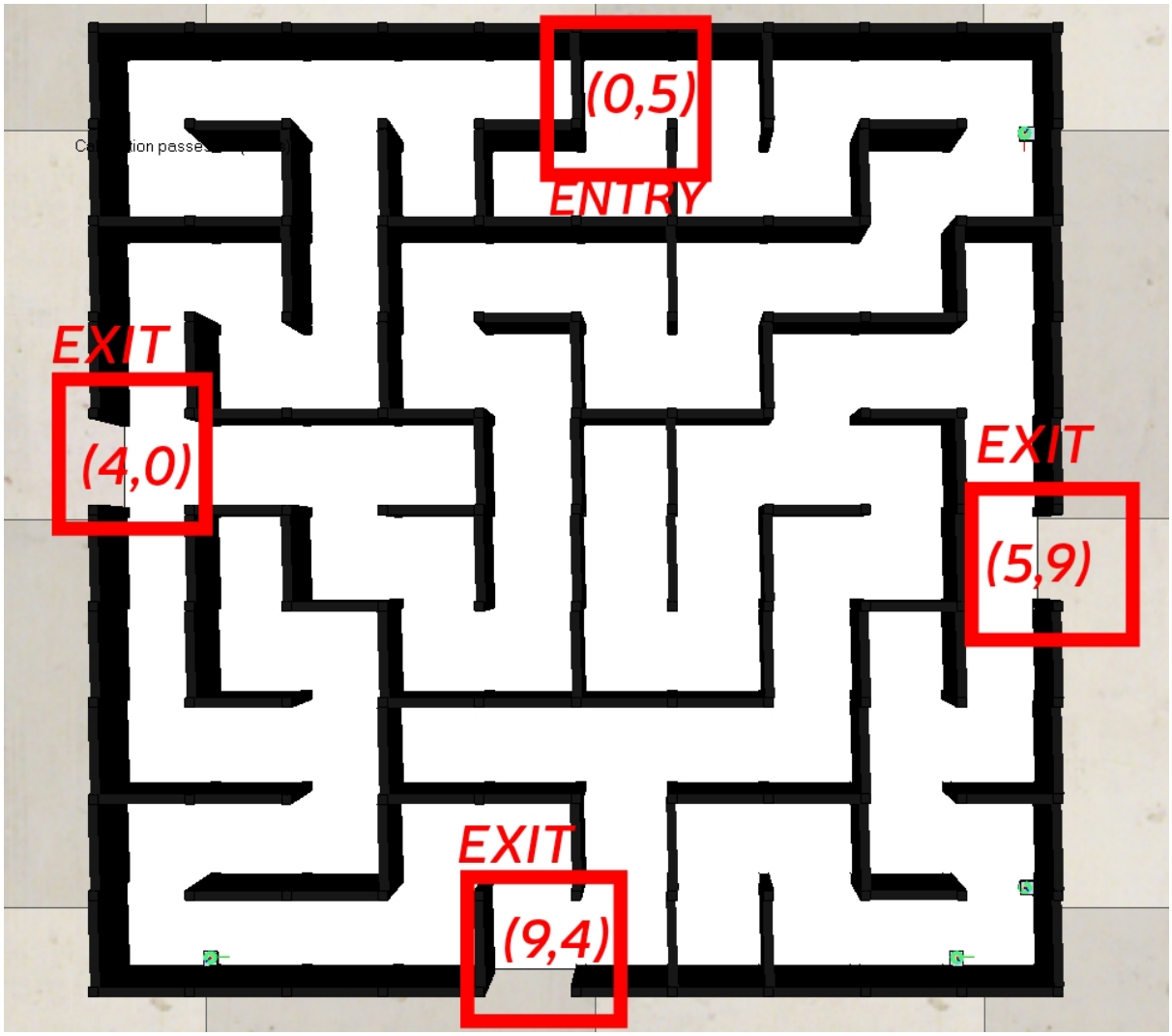
(II) maze_t2



(III) maze_t3



(IV) maze_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>

Answer:

Formula used for final score:

$$Score = \sum_{1}^{N} \Big[(CI * 10) + (CP * 100) + (CD * 50) + (CM_4 * CT_4 * 100) + (CP * CM_x * CT_x * 100) + (CI * CP * CM_4 * TB_4 * 10) + (CI * CD * CM_x * TB_x * 10) - (HP * 10) \Big] + [sum(CM) * 50]$$

For 1st ball in sequence (sequence according to Question 2) ie. 1st Green Ball

CI = 0 , CP = 1 , CD = 1 , CT_4 = 1 ,CT_x=CT_2=1, CM_4 = 1 , CM_x = CM_2 = 1 , TB_4 = 1 , TB_x= TB_2 = 1, HP=1

Putting values in summation formula we get =>

(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)

For 2nd ball in sequence (sequence according to Question 2) ie. 1st Blue Ball

CI = 1 , CP = 1 , CD = 1 , CT_4 = 1 , CT_x=CT_1=1, CM_4 = 1 , CM_x = CM_1 = 1 , TB_4 = 5 , TB_x=TB_1= 10, HP=8

Putting values in summation formula we get =>
(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 ……. (2)

For 3rd ball in sequence (sequence according to Question 2) ie. 1st Red Ball

CI = 0 , CP = 1 , CD = 0 , CT_4 = 1 ,CT_x=CT_3=1, CM_4 = 1 , CM_x = CM_3 = 0 , TB_4 = 3 , TB_x=TB_3=13, HP=4

Putting values in summation formula we get =>
(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 …….(3)

For 4th ball in sequence (sequence according to Question 2) ie. 2nd Green Ball

CI = 1 , CP = 0 , CD = 0 , CT_4 = 1 ,CT_x=CT_1=1, CM_4 = 1 , CM_x = CM_1= 1 , TB_4 = 10 , TB_x= TB_1 =20, HP=15

Putting values in summation formula we get =>
(1 * 10) + (0 * 100) + (0 * 50) + (1 * 1 * 100) + (0 * 1 * 1 * 100) + (1 * 0 * 1 * 10 *10) + (1 * 0 * 1 * 20 * 10) – (15*10)

=-40 …….(4)

For 5th ball in sequence (sequence according to Question 2) ie. 2nd Blue Ball

CI = 1 , CP = 1 , CD = 0 , CT_4 = 1 ,CT_x=CT_3=1, CM_4 = 1 , CM_x = CM_3= 0 , TB_4 =2, TB_x= TB_3 =17, HP=2

Putting values in summation formula we get =>
(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 .....(5)

Now summing up values (1) , (2),....(5) We get the value of summation as=>

340 + 430 + 160 + (-40) + 210 = 1100 .....(6)

Now in order to get final score we have to add [sum(CM) * 50] to (6)

where sum(CM) = CM_1 + CM_2+CM_3 + CM_4 = 1 + 1 + 0 + 1 = 3

[sum(CM) * 50] = 3 * 50 =150 .....(7)

Adding (6) and (7) We get the final score as =>

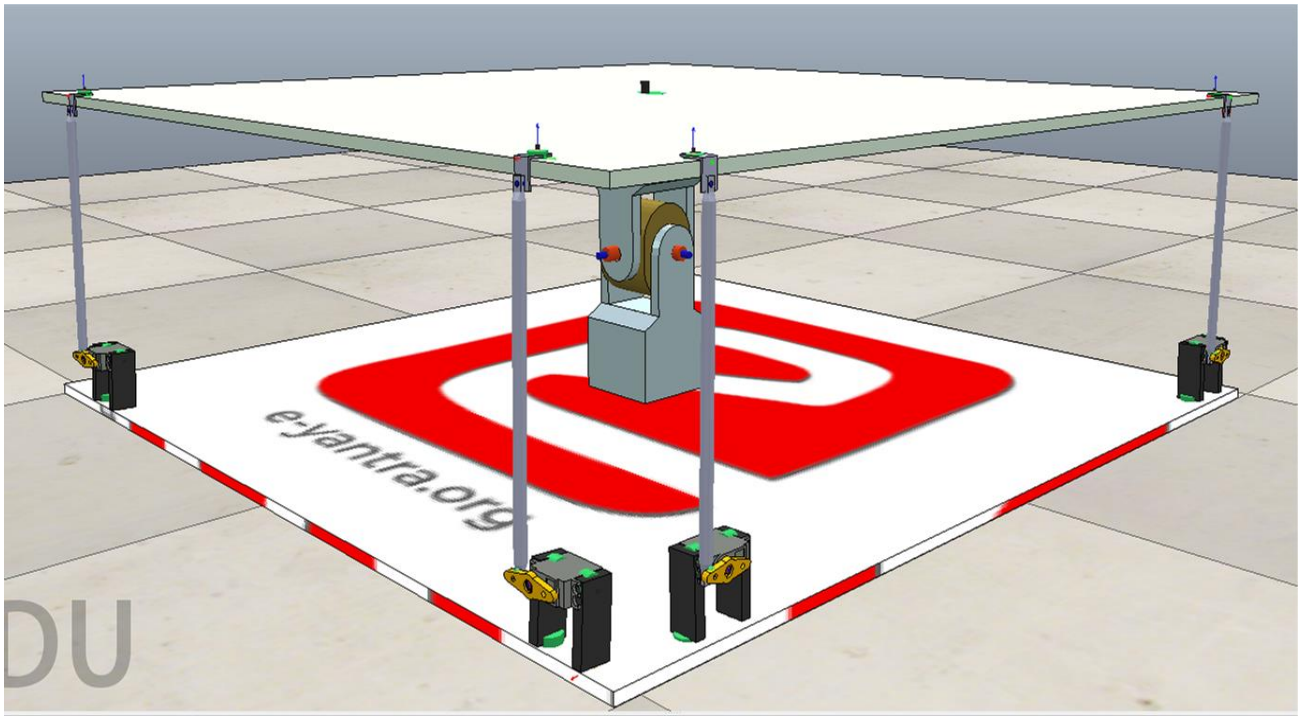
1100 + 150 = 1250

Mechanism

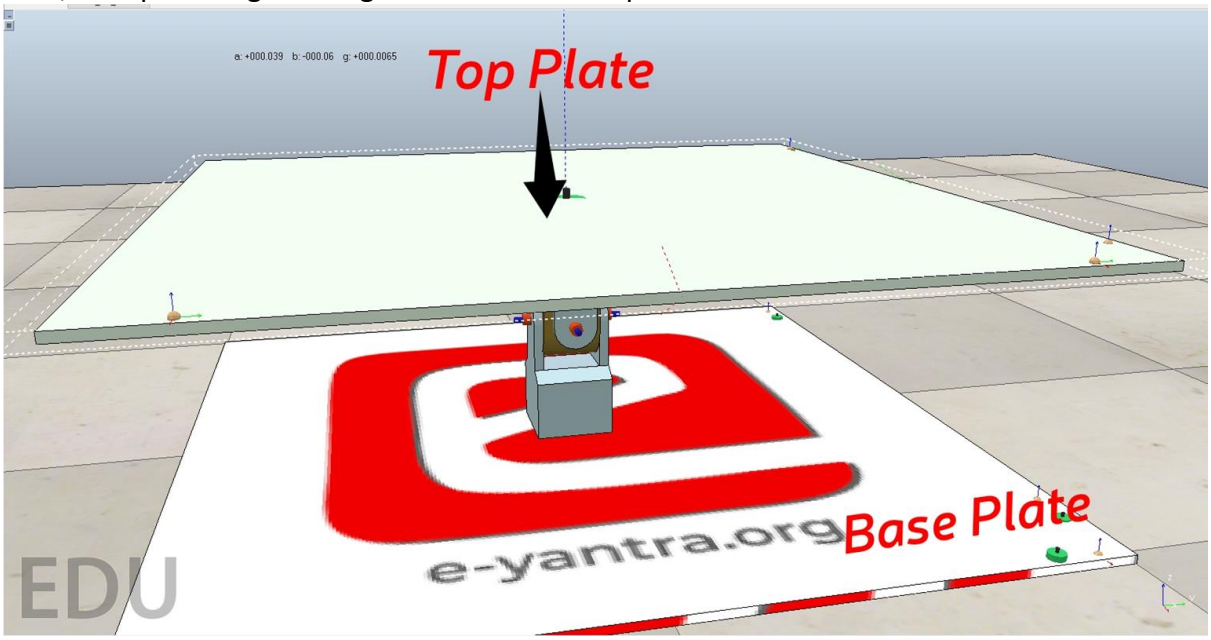
Q6. Explain the mechanism that you used for your ball balancing platform. (5)
< You must explain the mechanical construction of your ball balancing platform and how have you connected all the different components provided to you. Make properly labelled diagrams to show the same. You may also use screenshots of the CoppeliaSim scene to demonstrate your mechanism.>

Answer

Construction: The Ball balancing Platform consists of a base plate which is the base of the entire ball balancing system (and also the parent of all other components in coppelliaSim software.) Further more the ball balancing platform comprise of the following components.

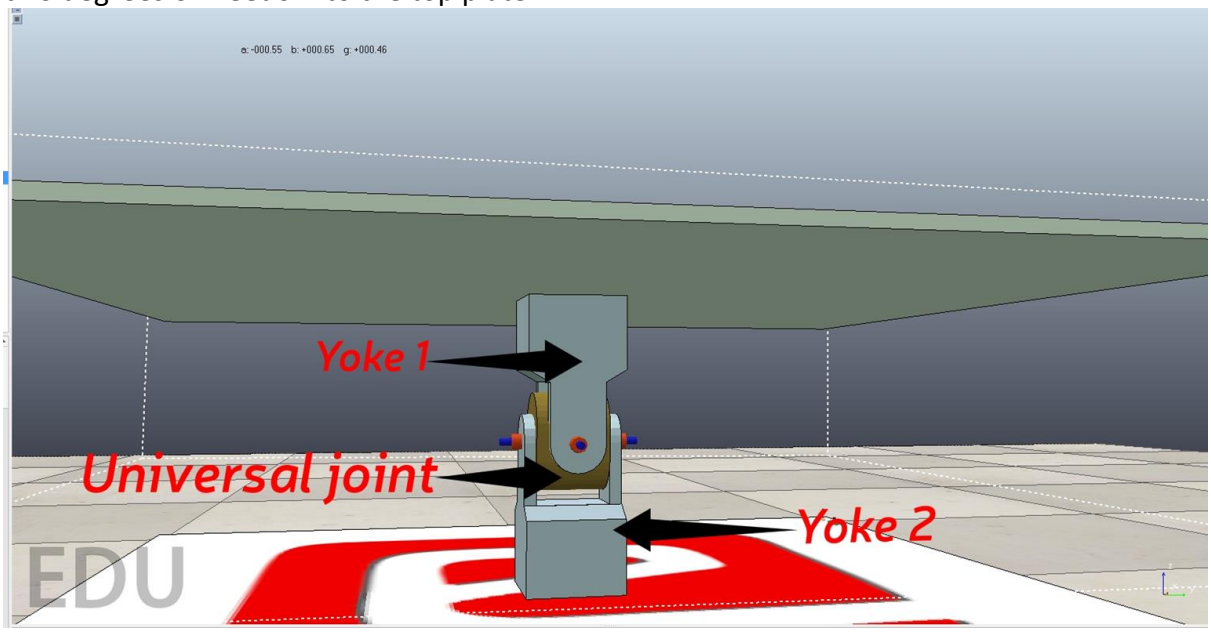


(I) Top Plate : It acts as the surface on which the ball is to be balanced . It is supported on its center by yoke universal joint support such that the plate is free to rotate about its x and y axis , thus providing two degrees of freedom to plate.



(II) YOKE-Universal Joint Support: There are two yokes used in this ball balancing platform .
Yoke1 is connected to top plate via means of a force sensor .
Yoke 2 is connected to bottom base plate via means of another force sensor.

Both the yokes are then connected to each other by means of universal joint and revolute joints such that the top yoke (Yoke 1) is able to rotate about both x and y axis. Thus providing two degrees of freedom to the top plate .

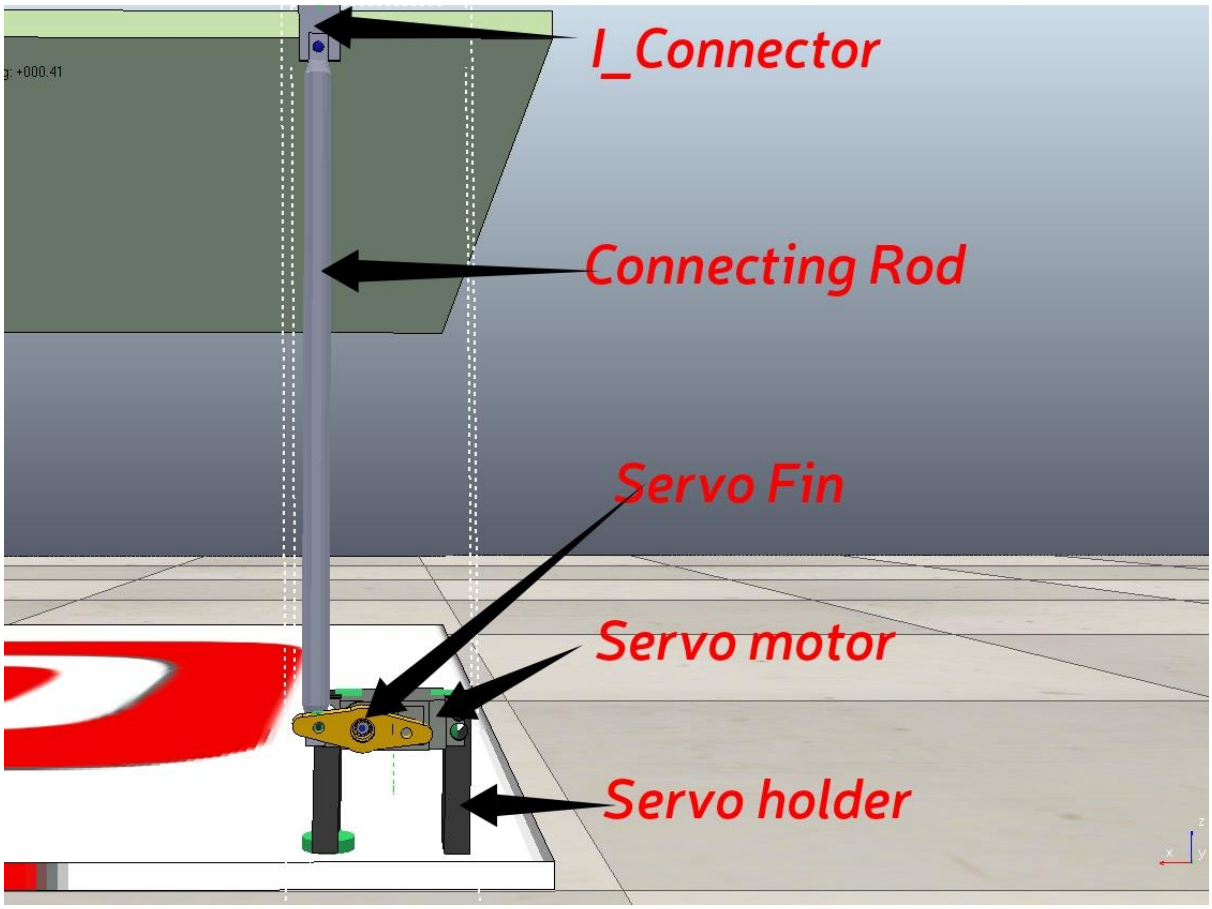


(III) Servo System: In a total 4 Servo Systems are used which act as actuator of the design .(Two each on the adjacent sides of the top plate). Each pair of servo systems which are attached to same edge of top plate are responsible for providing force to the top plate in the upward direction creating a torque in a direction parallel to the line joining those two servos (on the same edge).And thus the plate rotates due to the moment created by the force, about the axis of torque.

Each of the servo system comprise of the following :

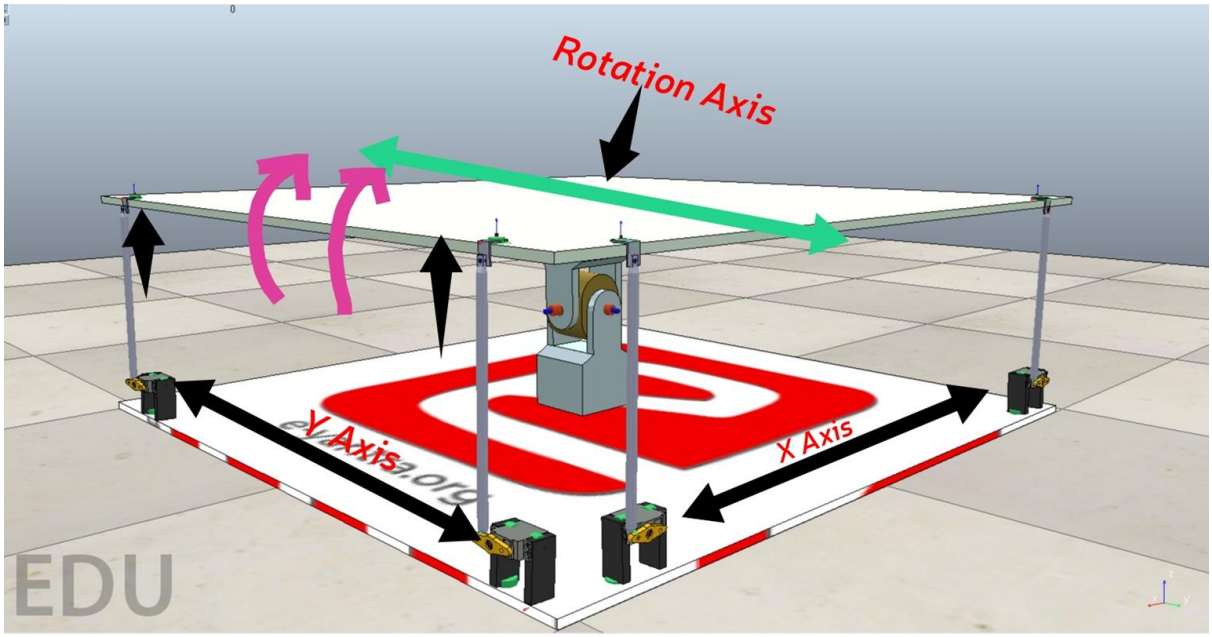
(a) ***Servo motor*** : It is the heart of the servo system and is responsible for rotation of the shaft which in turn rotates the fin connected to it and provide upward force to connecting rods.

(b) ***Servo Holder***: A pair of servo holders are used to support the the servo motor on either side and connect the servo system to the base plate.

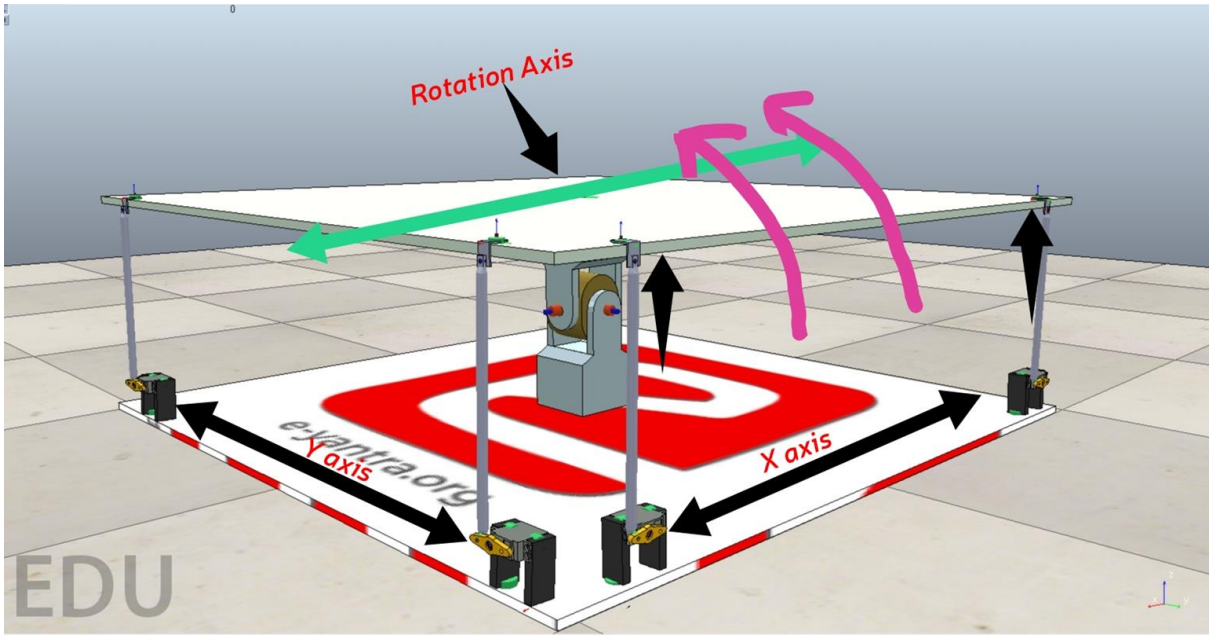


(c) ***Servo Fin***: It is connected to the shaft of the servo motor . When the shaft rotates the fin also rotates and with the upward or downward movement of fin the connecting rod also moves with it.

(d) ***Connecting Rod***: It is used to connect the servo fin to the top plate via some L-Connectors. Basically its job is to transfer the force generated by the shaft of the motor through the rotating servo fin to the top plate placed on YOKE-Universal Joint-YOKE support with 2 DOF , so that the plate could get a force rotating about either of the permitted axis.



(*when the servos on the Y-axis operate and servos rotate clockwise to provide upward force on the plate)*



(when the servos on the X-axis operate and servos rotate clockwise to provide upward force on the plate)

Similar will be the case when servo fins rotate in anti clockwise fashion just in that case the top plate will get a force in downward direction and the moment of force will be in clockwise manner.

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. .>

Answer:

Changes in Ball Balancing Design

=> The Servos in the design were a bit too close to center of the base plate , their position was changed and move to the border of the base plate , thus trying to maximise the distance between the points where the connecting rods will apply the upward push or downward pull and the center of base/top plate where the pivot for top plate ie. Yoke-Universal joint system was also present.

Reason:

Since Torque = Force * Perpendicular Distance between the point of application of force and the pivot

Now when we increased this perpendicular distance by moving servos to borders of base plate .. We had to apply less force to produce the same amount of torque..and this reduced force allowed us to control the angle of rotation of servo fin within the limits of -1.57 radians and 1.57 radians .

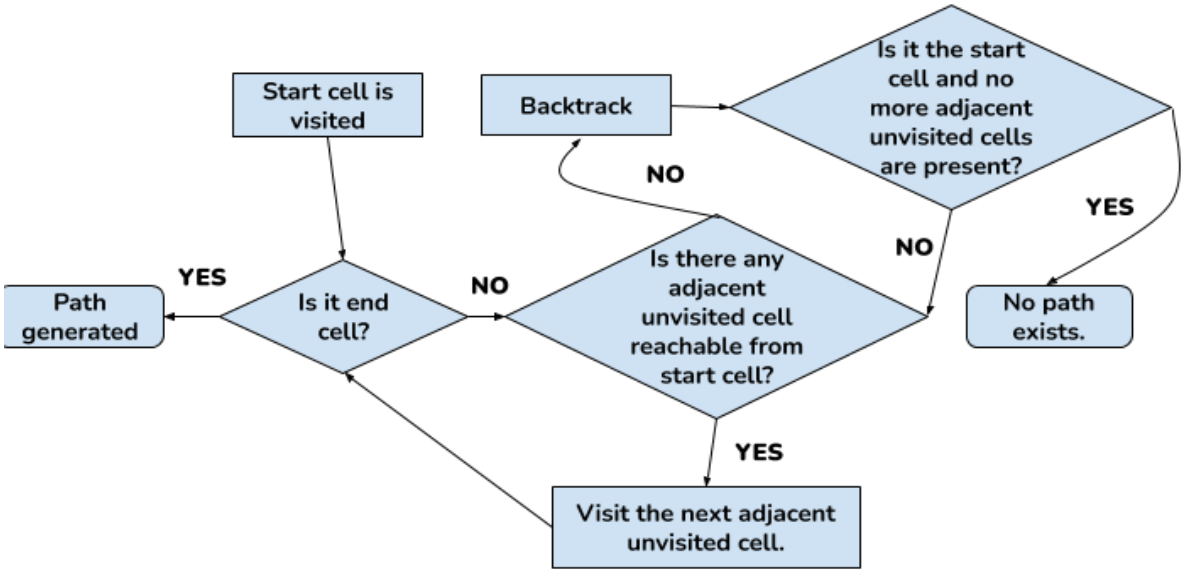
Path Planning

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. >

The path planning algorithm, we used in Task 4A, is **Depth-first search (DFS)**. The general reason for this choice is that DFS is easier to visualize for solving maze problems and takes into account only the path that is currently being pursued. And thus, uses lesser memory in most cases.

The logic behind this algorithm is defined as follows:



- The current cell (of the maze) in which the ball lies is marked as the visited cell, to avoid multiple processing of the same cell.
- If the current cell is the destination cell, the algorithm backtracks all the way to the starting cell, and the path is generated.
- If the current cell is not the destination cell, the following checks are made:
    - If the left wall doesn't exist, and the adjacent left cell is not visited, the ball is made to move to this cell.
    - If the top wall doesn't exist, and the adjacent top cell is not visited, the ball is made to move to this cell.
    - If the right wall doesn't exist, and the adjacent right cell is not visited, the ball is made to move to this cell.
    - If the bottom wall doesn't exist, and the adjacent bottom cell is not visited, the ball is made to move to this cell.
- The above steps are repeated until the current cell is the destination cell.
- If at any point, the ball cannot move to any cell from the current cell (due to presence of walls or all adjacent cells are already visited), backtracking takes place.

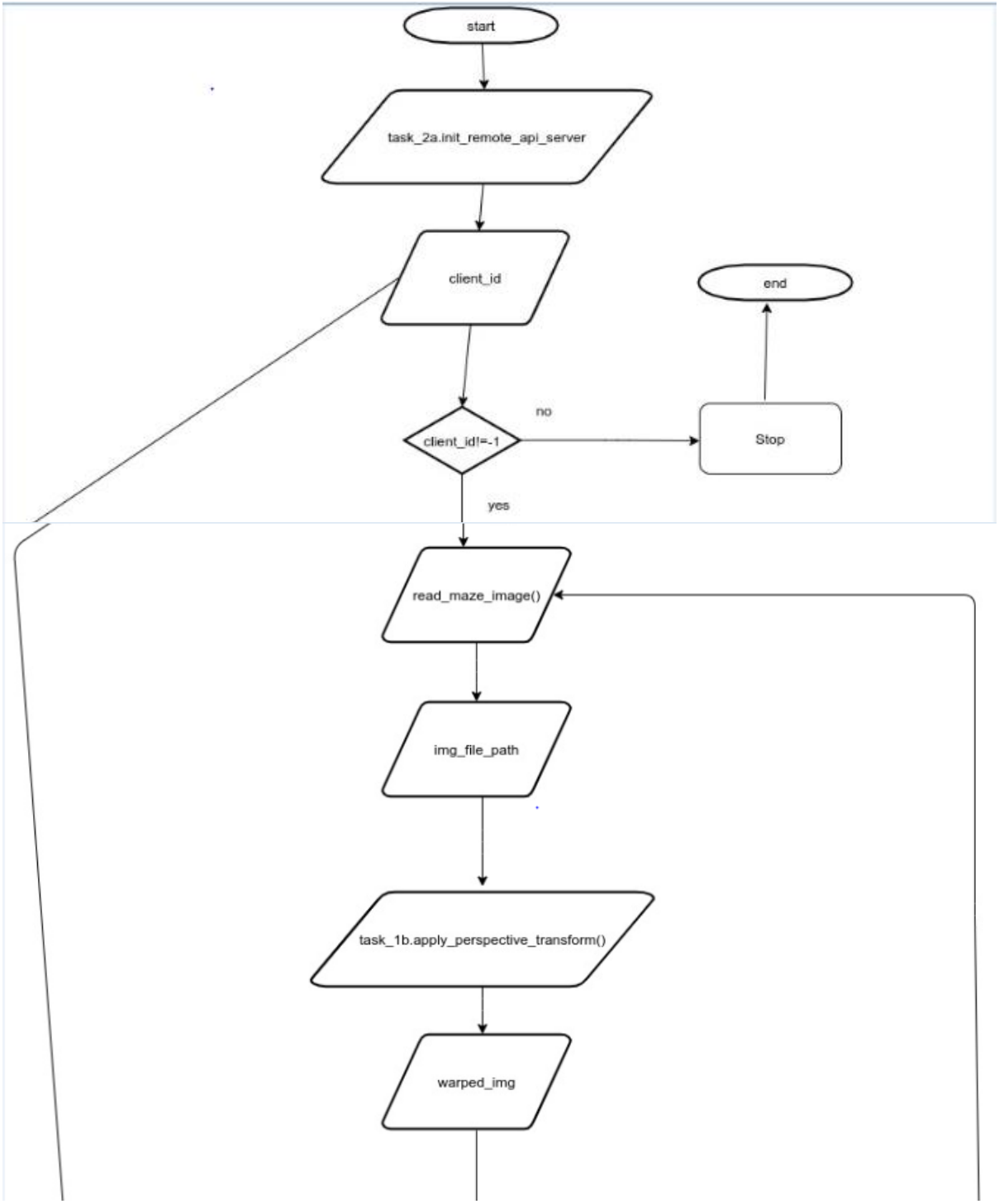
Algorithm Analysis

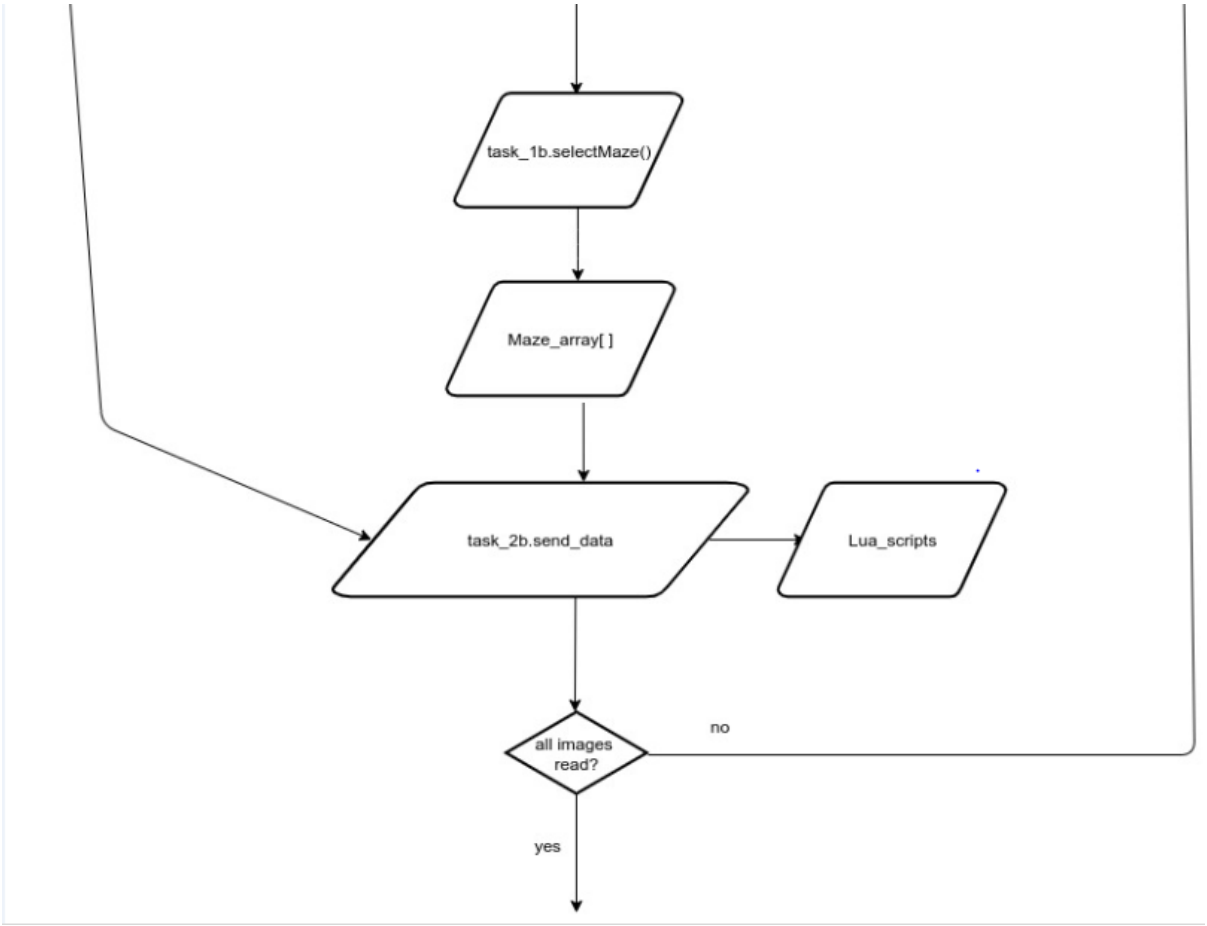
Q9. Draw a flowchart illustrating the algorithm / strategy you propose to use for theme implementation. (7)

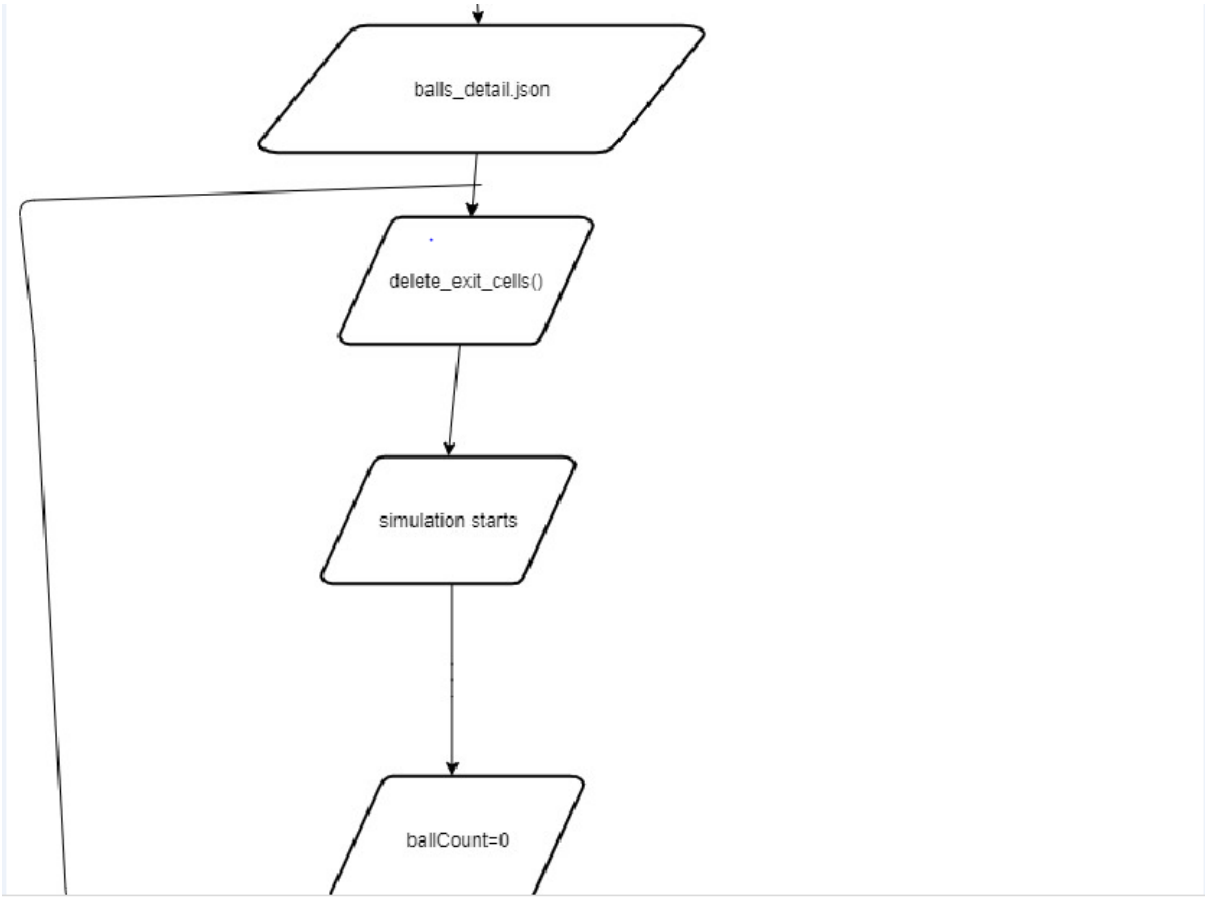
< The flowchart should elaborate on every possible function that you will be using for completing all the Theme Run.
Follow the standard pictorial representation used to draw the flowchart. >

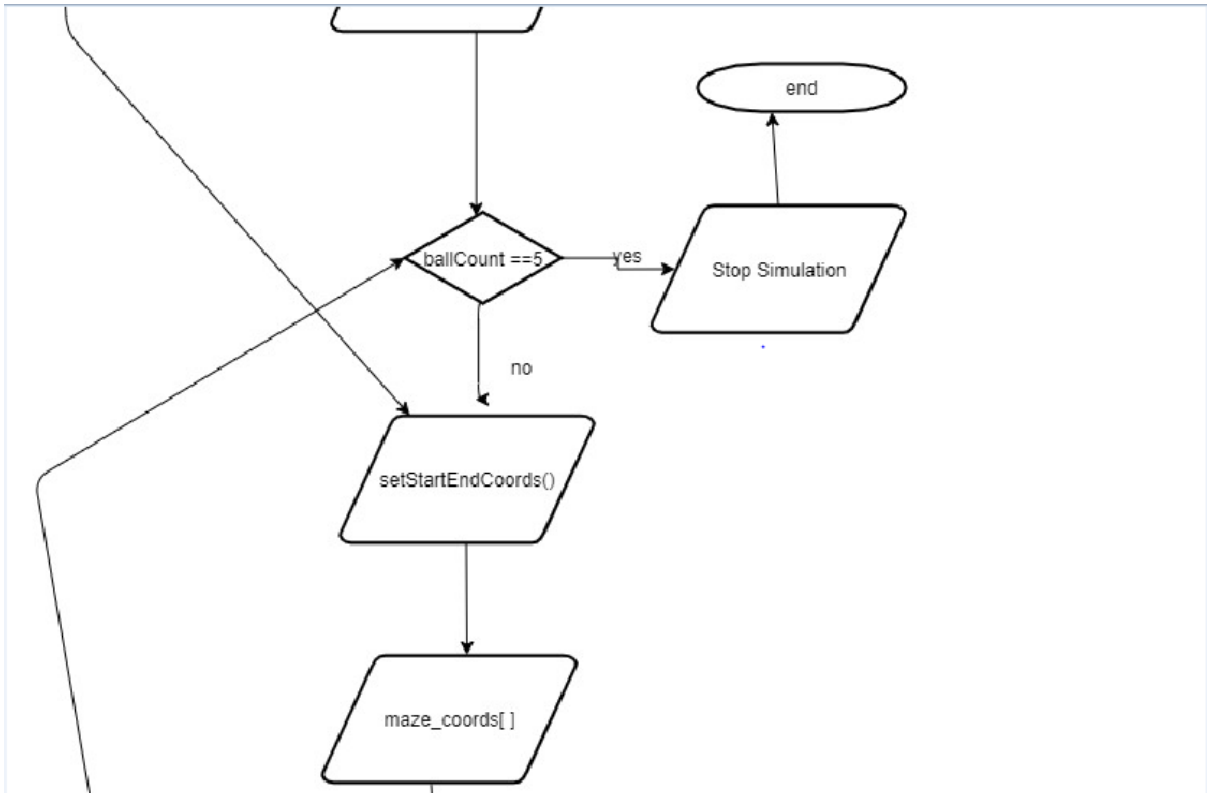
Flowchart for Implementation algorithm:

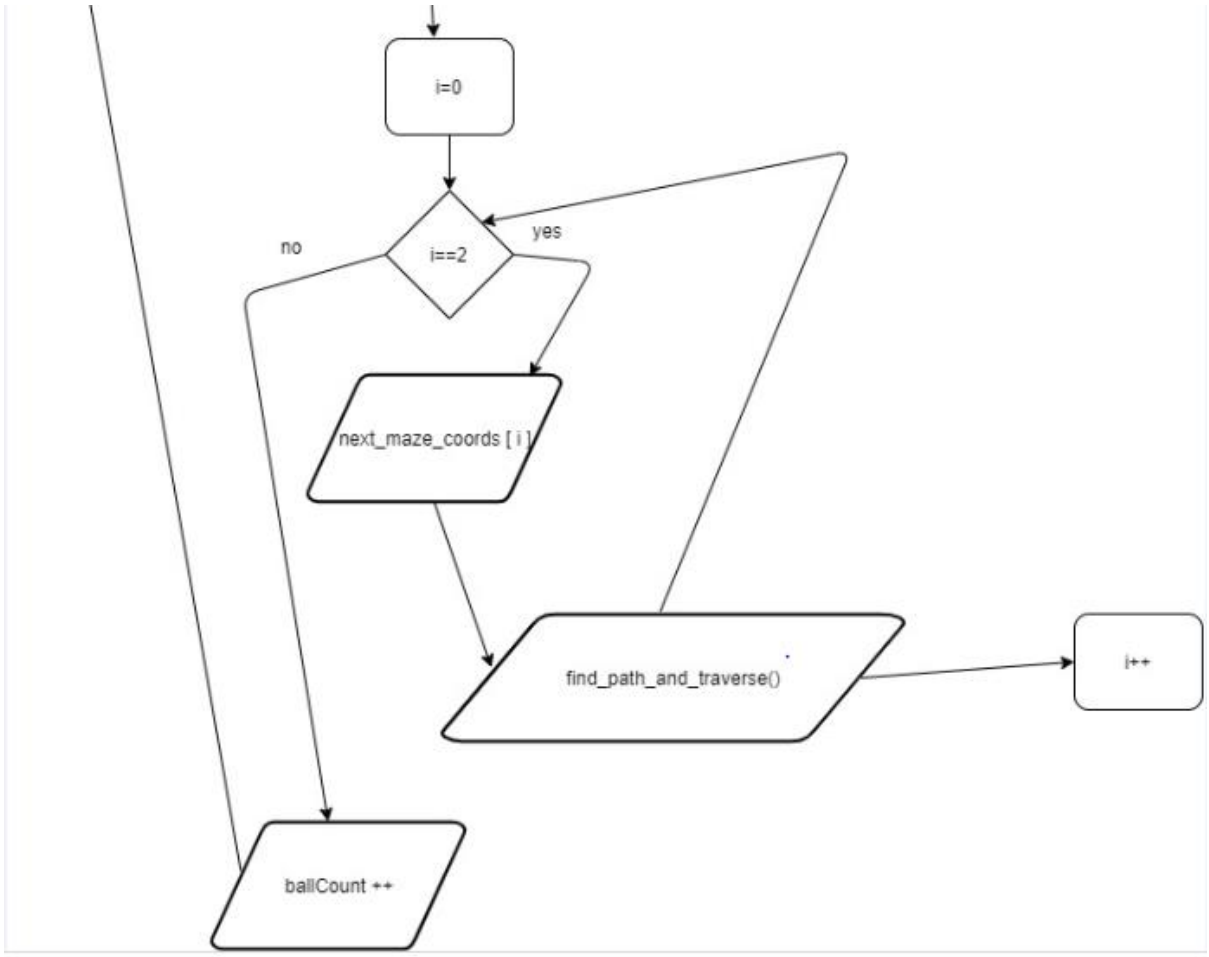
OBJ


start
task_2a.init_remote_api_server
client_id
end
no
client_id!=-1
Stop
yes
read_maze_image()
img_file_path
task_1b.apply_perspective_transform()
warped_img


task_1b.selectMaze()
Maze_array[ ]
task_2b.send_data
Lua_scripts
all images read?
no
yes


balls_detail.json
delete_exit_cells()
simulation starts
ballCount=0


end
ballCount ==5
yes
Stop Simulation
no
setStartEndCoords()
maze_coords[ ]


i=0
i==2
yes
no
next_maze_coords [i]
find_path_and_traverse()
i++
ballCount ++

Remarks:

1.The maze images are read one by one and the maze array generated is stored in global variable maze_array[] (list).

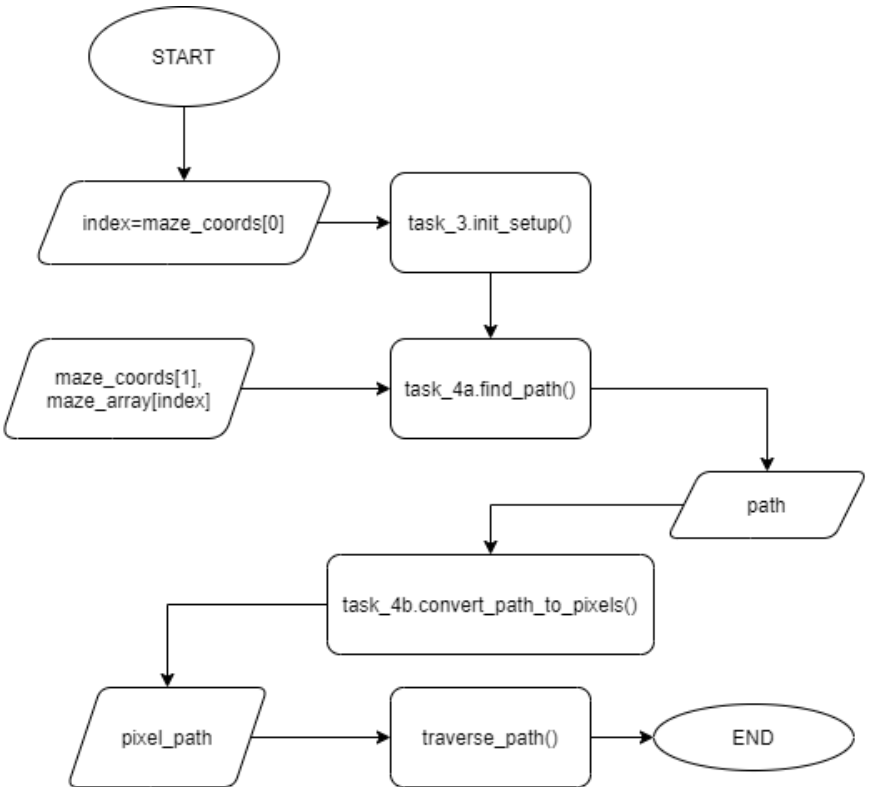
2.task_2b.send_data() sends the data to LUA functions to form the mazes on platform.

3. deleteExitWalls() function deletes the wall for the exit coordinates from all the mazes based on the ball_details.json file (identified by the color of the ball)

4. setStartEndCoordinates returns the index of the maze, and a list for the start and end coordinates, for both maze4 and the next maze to which the ball will go to.(setStartEndCoordinates returns a list with 2 elements). This is fed to find_path_and_traverse() which further finds path and traverse ball.

5. find_path_and_traverse() is run twice, first for the 0th element, i.e for maze4 and then for the next maze where the ball will go.

Flowchart for function find_traverse():



Remarks:

1. Here index,i.e. the maze index is determined by the 0th element of the argument and the next element determines the start and end coordinates for the ball in the maze.

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. >

1. We faced errors in detecting the required contours including detection of noise as contours in vision-sensor-captured images, and sometimes detection of no contour at all which is attributed to the fact that there is no specific order of contours being detected.
2. It was particularly difficult to work with image processing initially due to resulting flipped or rotated warped images.
3. It took a considerable amount of time to fine tune almost every constant used in the tasks such as in thresholding, masking, PID constants etc. A lot of hits and trials went into it to narrow them down to suitable values.
4. Improving the “Real Time Factor” in CoppeliaSim was one of the major changes as the remote API commands penalized our client. Transition from synchronous mode to asynchronous one was a long journey resulting in a highly improved real time factor and more responsiveness of our control logic.
5. Intricacies involved in finding the optimum positions of actuators, universal joints, and other components in the ball balancing platform.
6. The challenges that we anticipate for future tasks are mainly concerned with balancing the ball on multiple platforms leading to synchronization caveats and handling multiple PID systems at the same time which may lead to increased lag and thus a reduced real time factor.

We plan to resolve this by using the operation modes and socket communication between the server and the client in an efficient manner.

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.