



e-Yantra Robotics Competition - 2020-21

Nirikshak Bot

Task 4C - Theme and Implementation Analysis

2182

Team Leader Name	Pranav Mittal
College	IIT(BHU)
Team Leader Email	2001pranavmittal@gmail.com
Date	13 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. >

Our theme is self balancing platform that gives us the idea about control systems , they can be very useful in travel industry as in our today modern age drones and robots will be next and faster travel options.

For sensitive goods we want them to be stabilized, self balancing platforms can be useful in balancing goods on the platform so that they don't fall down.

Or a more commercial approach would be to make the everyday commute of everyone more genuine like cars, trains that have self balancing base that is not affected by the angle the vehicle is travelling. In that case the one of prime source of error would be gyroscope.

Or In a more general sense the control system such as PID will give us a start at more complex projects such as collision resistant drones that will take data from ultrasonic sensors and ultimately slow down as required.

Or simply drone travel can be implemented using same way using GPS systems as well.

As PID is a general concept and provides a simple solution in infinite real world problems.

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.

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

A)FIRST GREEN BALL

T4 ENTRY(0,5): EXIT(9,4)→T2 ENTRY(0,4):EXIT(9,5)→T2_CB2

B)SECOND BLUE BALL

T4 ENTRY(0,5): EXIT(4,0)→T3 ENTRY(4,9):EXIT(0,4)→T3_CB3

C)FIRST RED BALL

T4 ENTRY(0,5): EXIT(4,0)→T3 ENTRY(4,9):EXIT(9,5)→T3_CB1

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>

The walls for exit has been deleted by using a function called deleteExit(number) in lua , that takes the table number as inout and deletes the walls accordingly.

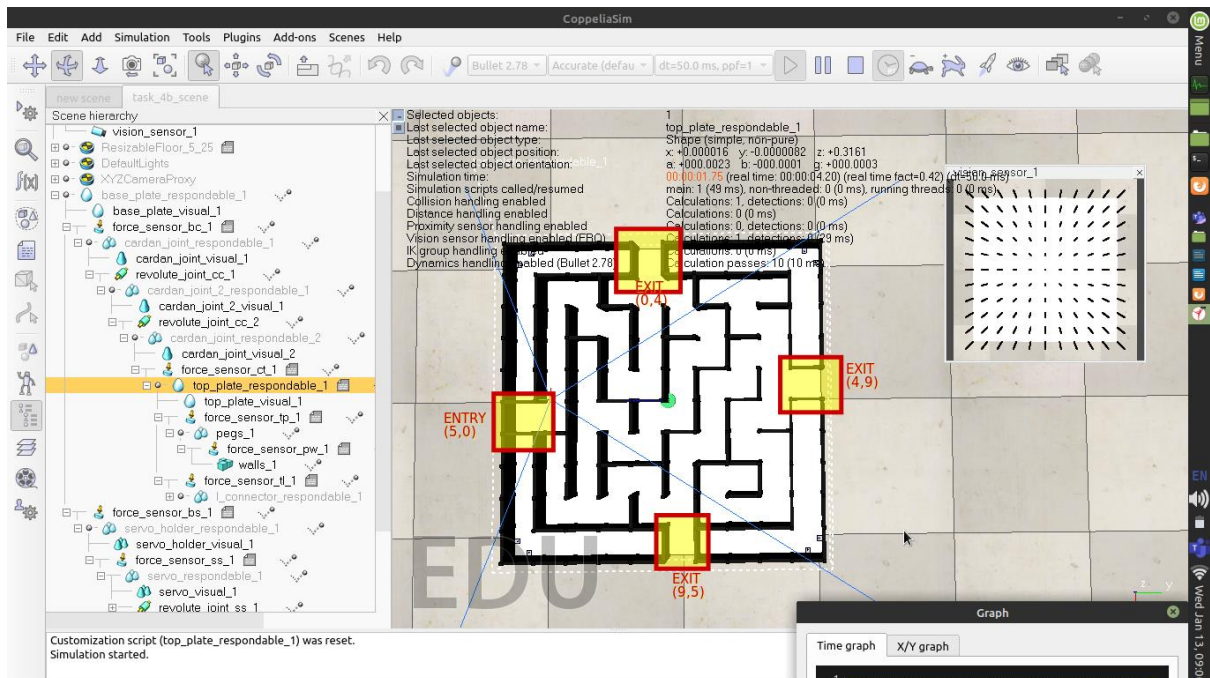
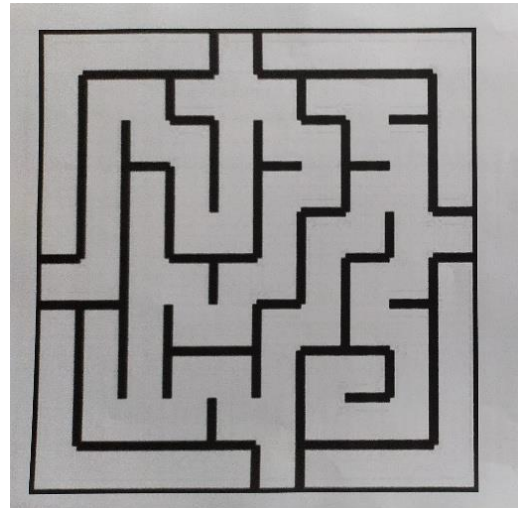
function deleteExit(number)

```
--wrong function need debugging
--number is index of table
--deleting extra walls according to table number
--for Table4(Center),Exit points are Vertical_wall-(4,0),vertical_wall-(5,9),horizontal_wall-
(9,4)
--for Table1(Right),Exit points are Horizontal_wall-(0,4),vertical_wall-(4,9),horizontal_wall-
(9,5)
--for Table2(Bottom),Exit points are Vertical_wall-(5,0),vertical_wall-(4,9),horizontal_wall-
(9,5)
--for Table3(Left),Exit points are Vertical_wall-(5,0),horizontal_wall-(0,4),horizontal_wall-
(9,5)
--the names of walls start from one where as wall number start from 1
--FORMAT:list[table][i][1]==1,remove    horizontal    else    remove    veritcal    at
list[table][i][2]xlist[table][i][3]
if(number>4) or (number<1)
then
    return
end
list={{1,1,5},{-1,5,11},{1,11,6}},
      {{-1,6,1},{-1,5,11},{1,11,6}},
      {{-1,6,1},{1,1,5},{1,11,6}},
      {{-1,5,1},{-1,6,11},{1,11,5}}}
for i=1,3,1
do
    --print(i)
    --print(list[number][i])
    if(list[number][i][1]==1)
    then
        name="H_WallSegment_"..list[number][i][2].."x"..list[number][i][3]
        sim.removeObject(sim.getObjectHandle(name))
    else
        name="V_WallSegment_"..list[number][i][2].."x"..list[number][i][3]
        sim.removeObject(sim.getObjectHandle(name))
    end
end
end
end
=====
```

Encoded Maze-Array

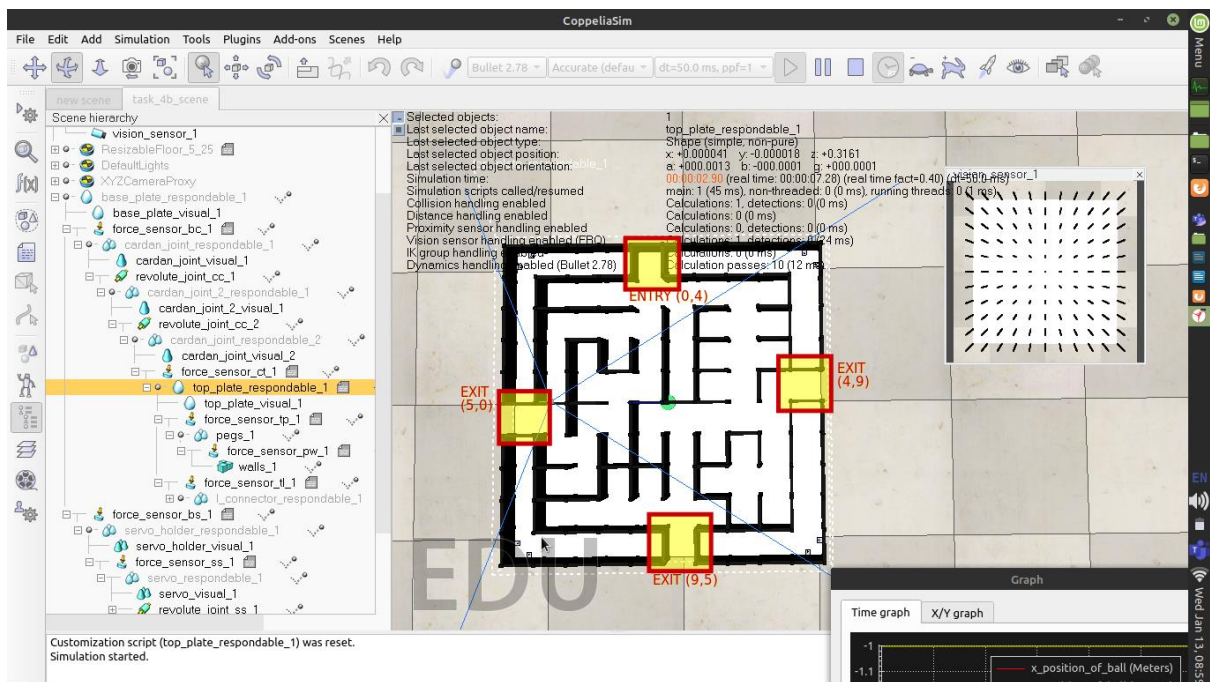
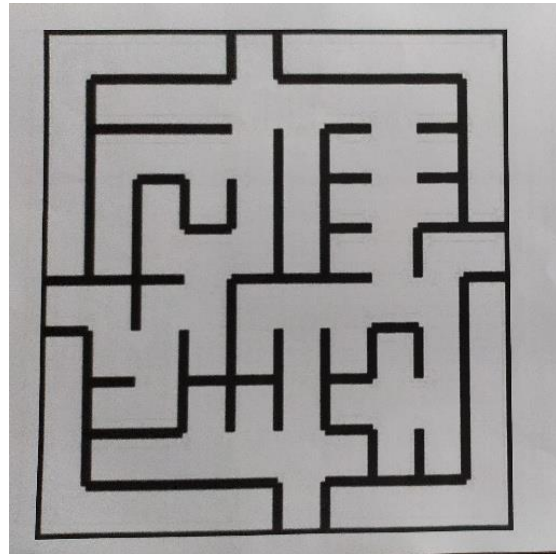
t1

```
[[3, 10, 10, 14, 7, 11, 10, 10, 10, 6],
 [5, 3, 6, 11, 0, 6, 11, 2, 14, 5],
 [5, 5, 9, 6, 5, 9, 6, 9, 6, 5],
 [5, 5, 7, 5, 5, 3, 12, 3, 4, 13],
 [13, 5, 5, 9, 12, 5, 3, 12, 1, 14],
 [11, 12, 1, 6, 3, 12, 5, 3, 12, 7],
 [7, 7, 5, 9, 12, 3, 12, 9, 6, 5],
 [5, 5, 5, 3, 6, 5, 3, 14, 5, 5],
 [5, 9, 8, 12, 9, 4, 9, 10, 12, 5],
 [9, 10, 10, 10, 14, 13, 11, 10, 10,
 12]]
```



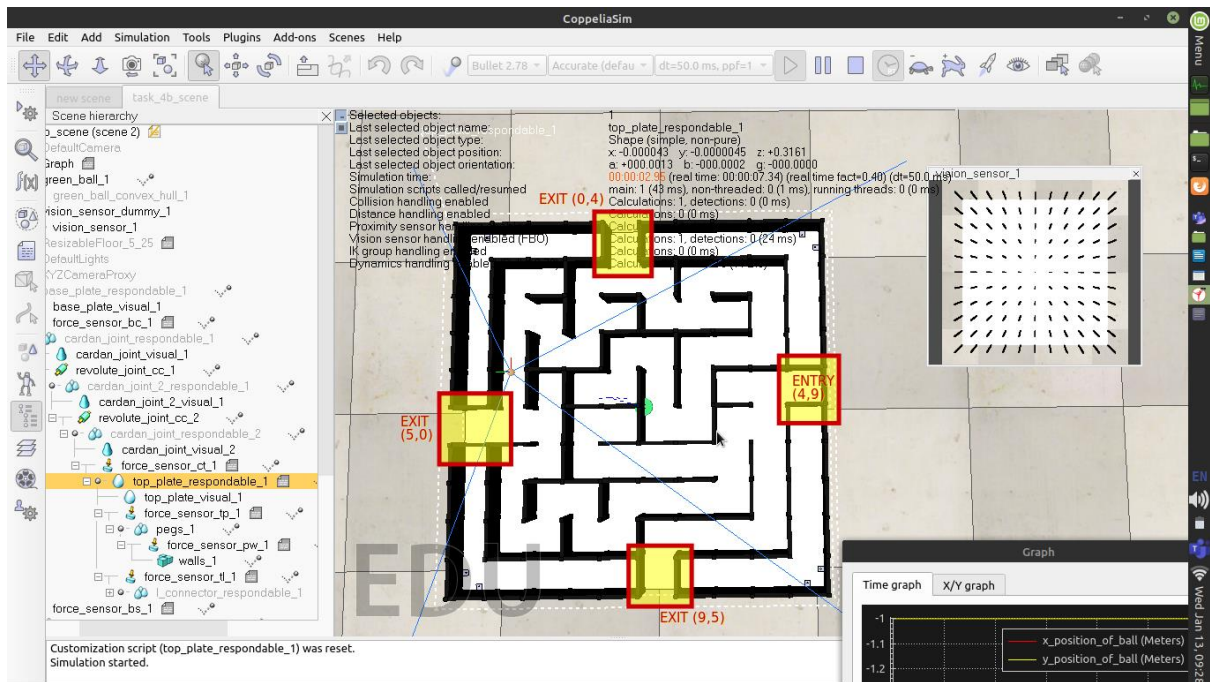
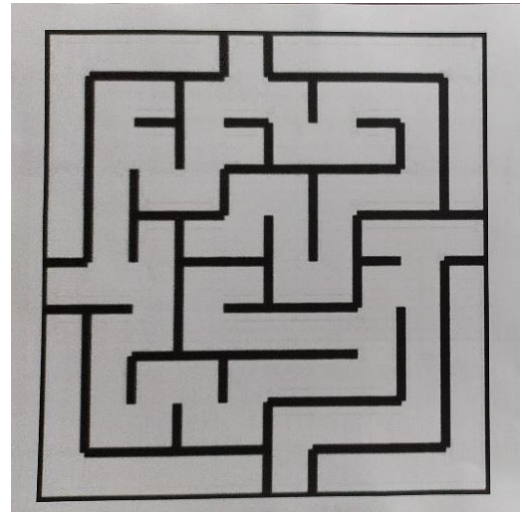
t2

```
[[3, 10, 10, 14, 7, 11, 10, 10, 10, 6],
 [5, 11, 10, 10, 0, 2, 10, 2, 14,
 5],
 [5, 3, 10, 2, 4, 5, 11, 0, 14, 5],
 [5, 5, 7, 13, 5, 5, 11, 0, 14, 13],
 [13, 13, 9, 2, 12, 13, 11, 4, 3,
 14],
 [11, 6, 3, 4, 3, 2, 2, 8, 4, 7],
 [7, 9, 4, 13, 13, 5, 13, 7, 5, 5],
 [5, 11, 12, 7, 7, 5, 11, 0, 4, 5],
 [5, 11, 10, 8, 8, 0, 14, 13, 13,
 5],
 [9, 10, 10, 10, 14, 13, 11, 10, 10,
 12]]
```

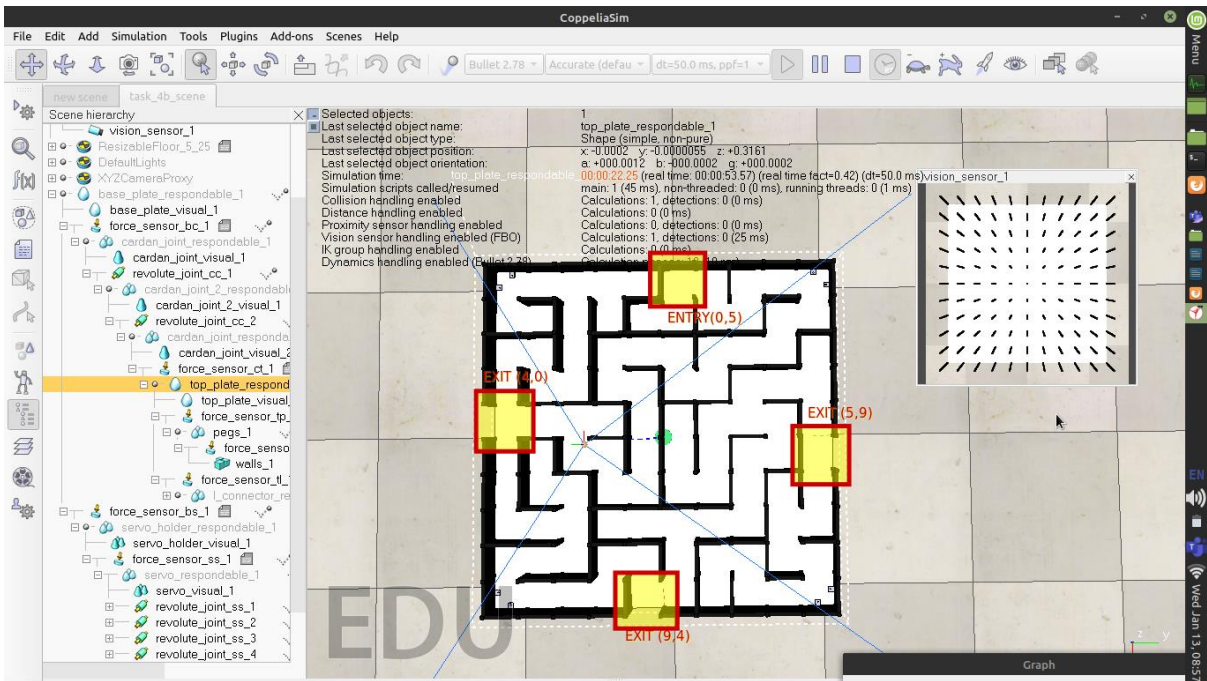


t3

```
[[3, 10, 10, 14, 7, 11, 10, 10, 10, 6],
 [5, 3, 14, 3, 8, 6, 3, 10, 6, 5],
 [5, 1, 6, 1, 14, 9, 8, 14, 5, 5],
 [5, 5, 9, 12, 3, 6, 3, 10, 12, 13],
 [13, 5, 7, 11, 12, 5, 5, 11, 2,
 14],
 [11, 8, 4, 3, 14, 9, 12, 3, 4, 7],
 [7, 3, 12, 9, 10, 10, 10, 4, 5, 5],
 [5, 5, 3, 6, 3, 10, 10, 12, 5, 5],
 [5, 9, 12, 9, 12, 3, 10, 10, 12,
 5],
 [9, 10, 10, 10, 14, 13, 11, 10, 10,
 12]]
```




```
[[3, 10, 2, 2, 14, 3, 6, 3, 10, 6],
 [9, 14, 5, 13, 11, 12, 9, 12, 3,
 12],
 [3, 6, 5, 3, 10, 6, 3, 10, 12, 7],
 [5, 9, 12, 9, 6, 9, 12, 3, 10, 4],
 [1, 10, 2, 14, 5, 7, 3, 8, 6, 5],
 [5, 7, 9, 6, 5, 5, 5, 3, 12, 5],
 [5, 9, 6, 9, 12, 9, 12, 5, 7, 5],
 [9, 10, 4, 11, 10, 2, 10, 12, 1,
 12],
 [3, 10, 12, 3, 6, 5, 3, 6, 9, 6],
 [9, 10, 10, 12, 9, 12, 13, 9, 10,
 12]]
```



	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>

Score= $\sum 1N[(CI*10)+(CP*100)+(CD*50)+(CM4*CT4*100)+(CP*CMx*CTx*100)+(CI*CP*CM4*TB4*10)+(CI*CD*CMx*TBx*10)-(HP*10)]+[sum(CM)*50]$

From Q2 the drop box of each ball according to balls_details.json is

Sequence	Color	Collection Box Name	CMx
4th	Green	T1_CB1	1
5th	Blue	T3_CB3	0
2nd	Blue	T1_CB3	1
3rd	Red	T3_CB1	0
1st	Green	T2_CB2	1

	CI*10	CP*100	CD*50	CM4*CT4*100	CP*CMx*CTx*100	CI*CP*CM4*TB4*10	CI*CD*CMx*TBx*10	HP*10	T
1	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
2	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
3	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
4	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
5	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

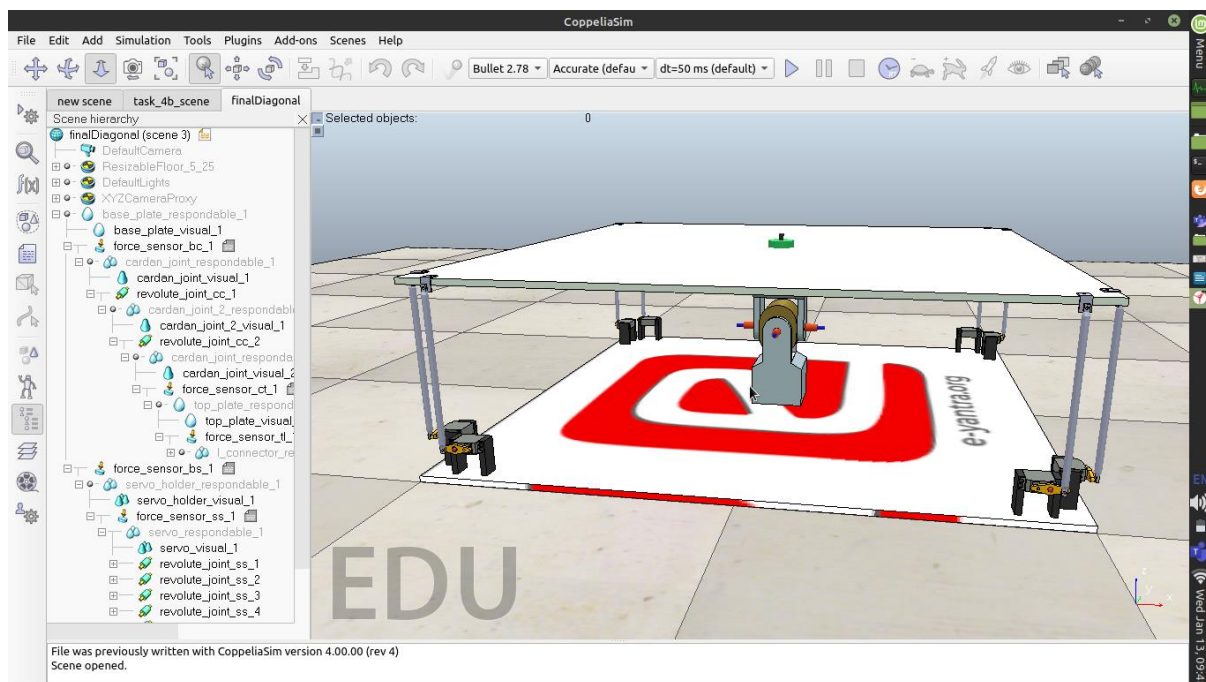
$$\text{Total} = (340 + 430 + 160 - 40 + 210) + [(1 + 1 + 0 + 1) * 50] = 1100 + 150 = 1250$$

Mechanism

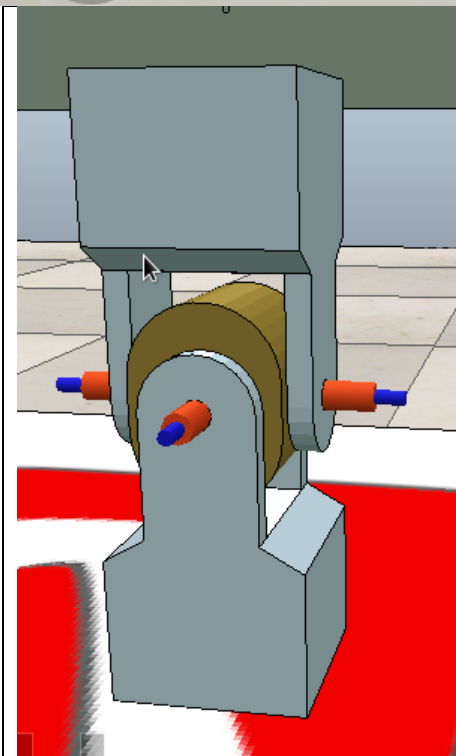
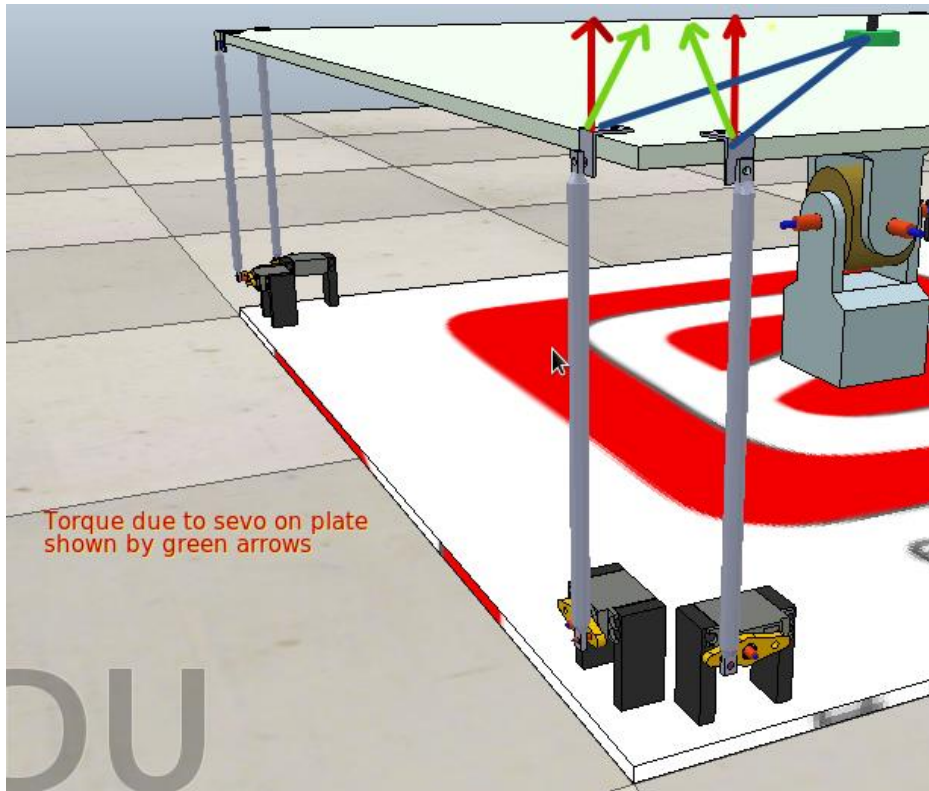
Q6. Explain the mechanism that you used for your ball balancing platform. (5)

< 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.>

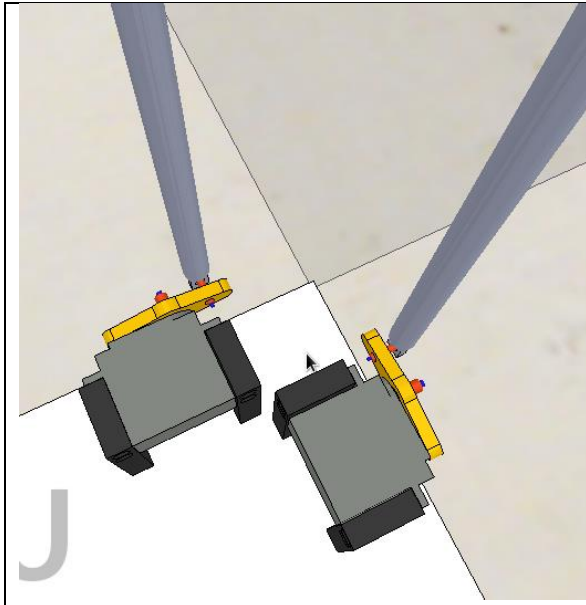
Basic: servo's are connected on the base move the top plate , so that the ball moves to the desired position.



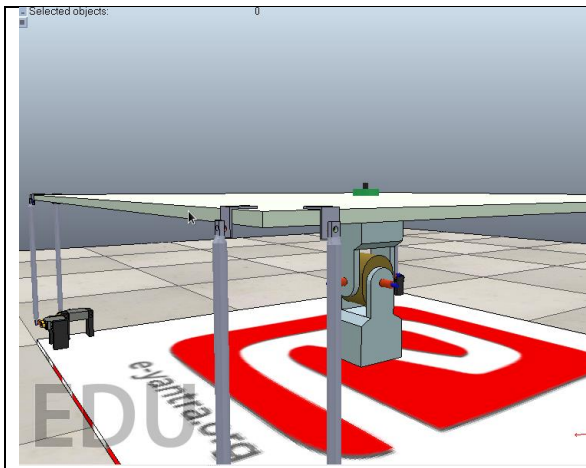
The base plate is connected to the servo's which are symmetrically placed , to ensure that there are minimum vibration and most accuracy when an angle is assigned to the them . The symmetric counterpart of a servo prevents the unprecedented movement of top plate along their respective axis of rotation.



The yoke support the top plate's weight ,It has been implemented using two revolute joints placed perpendicular to each other that allow for movement of top plate in both X-Y directions thus allowing to freely balance the ball.



The servo is connected to servo fin using a revolute joint that allows for 360deg rotation along 1 axis (but we have constrained it to -90 to 90 in code), which is further connected to connecting rod that allows for 360 deg rotation along 1 axis, this is how the servo is able to propagate the angle to top plate via connecting rod.



The connecting rod and I connector are connected via a spherical joint that allows **360 degree** rotation along all the axis, this allows the connecting rod to move as a part of top plate and decreases the strain on the joint (which would have been very high in case of revolute joint which won't allow the I connector to move both along x and y axis).

The other components have been connected, as if they were glued/fixed together. The paths are all I connector and top plate, cardan joint and top plate, cardan joint and base plate, servo holder and base plate, servo holder and servo.

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. .>

The location of servo's is very significant in order to implement the algorithm correctly. During task1c the main focus was to make a model that is very stable and minimum vibrations .

From task3 the main focus changed to make the ball easily *move* so the position of servo was changed to other location where the output given by pid algorithm shows accurate effect on the movement of the ball.

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 was calculated using searching through the maze for possible paths kindof like tree traversal. We can term this traversal similar to depth first search.

PsuedoCode

1.Creation of distance matrix, it stores the distance of traversable points form the start_coord
We start from start_coord and assign its distance=1,

k=1

while(we dont reach end_coord){

 -We search for point where distance = k

 loop{

 Now we start looping to all the neighbouring points of that node that have distance value in matrix =0 and the points where maze_array allows us

 set it to k+1

 }

 k=k+1

}

by doing this we get a matrix that have distance is the increasing order and when we reach the end_coord the value of k is the minimum distance that has to be travelled.

Now to find path (Recursion is used)

We start from the end_coord using the distance matrix we loop back through all the possible paths and keep on discarding the options on the way

if we reach the start_coord we save that path

RESULT: by doing this we can get multiple paths to the same points.

Checking which path is the best

Here we check this with the perspective of pid, For fast implementation of pid we need to see which path has minimum number of corners as ball needs to be slow around them.

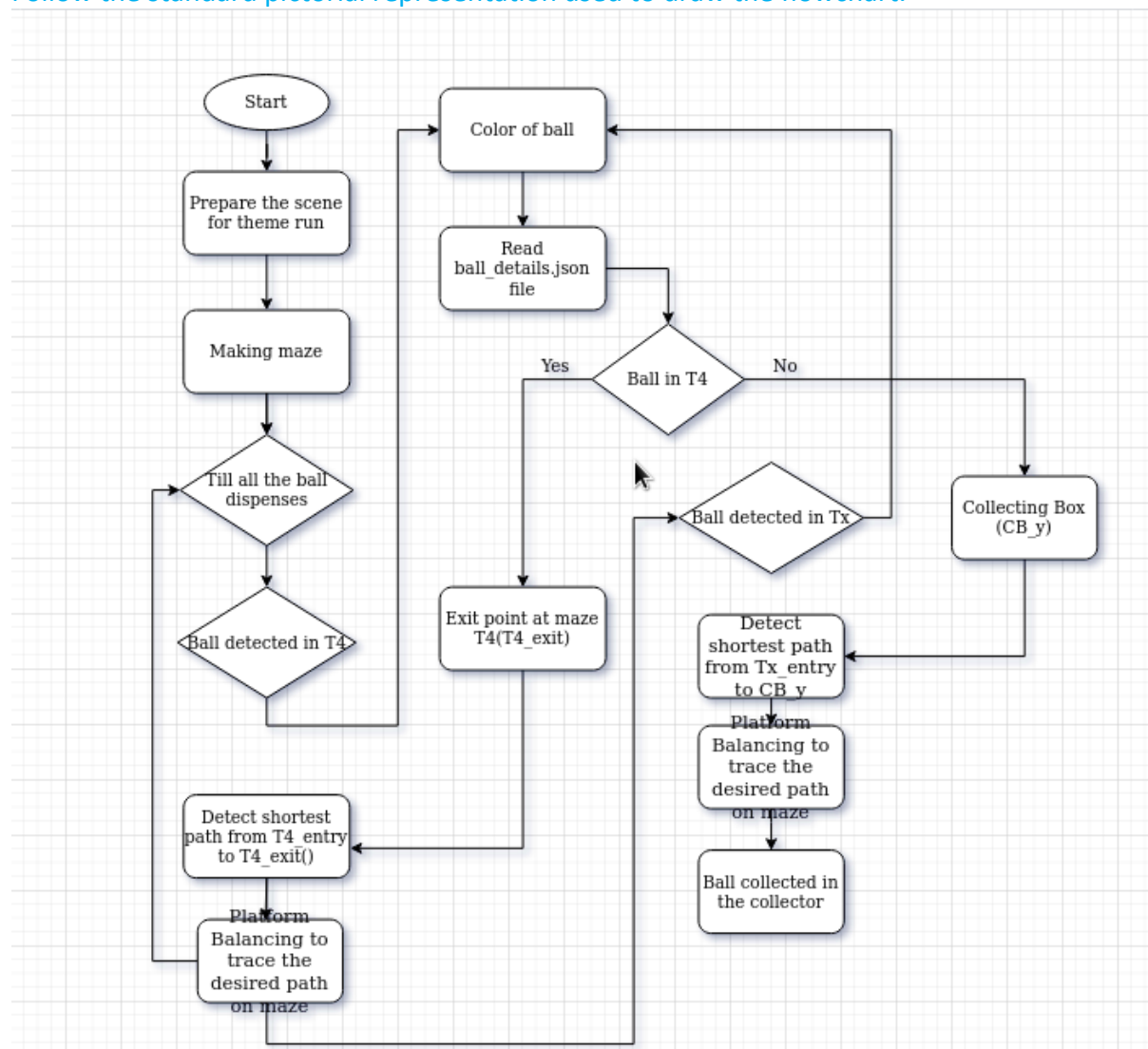
So we loop through all the possible paths that we got and see which path has minimum corners using the property that at least 3 consecutive points will have same x or y value . The redundant points (not the corners) are removed.
We return the path that has minimum corners.

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. >



Functions used in ::

Making maze:

receiveData(), generateHorizontalWalls(), generateVerticalWalls(), deleteWalls(), createMaze(), deleteExit(number)-"delete walls based on table number"

Ball Detection in T4:

scan_image(), get_vision_sensor_image(), transform_vision_sensor_image(), applyPerspectiveTransform(transformed_image)

Detect shortest path between T4_entry to T4_exit:

applyPerspectiveTransform(), detectMaze(), read_start_end_coordinates(), find_path()

Platform balancing:

control_logic(center_x,center_y), change_setpoint()

Ball detected in Tx:

decideExitPoint(ballDetails,tableNumber)

Detect shortest path between Tx_entry to CB_y:

applyPerspectiveTransform(), detectMaze(), read_start_end_coordinates(), find_path()

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.Debugging

As we kept on progressing with the task the code started to become very long and debugging started to become very difficult. So we decided to keep all the comments in code and start keep long meets to catch error faster.

2.Model

There was constant need to make the ball to move accurately, the position of servo's had to be changed again and again which took time and understand coppeliasim , how to make connections. The reviews in the result and piazza post helped in understanding what to do.

3.Team Collaboration

In our team not everyone has the perfect laptop to execute the simulation perfectly and as pid tuning takes a lot of it makes very much burden on one person to do so. Also if there is error in the code it takes double the normal time it would have. It is difficult to understand someone's code without their presence that fast.

4.Figuring out correct thresholding

It was very difficult to find a correct method to threshold such that ball is detected , maze is detected, if both are there then we have to separate ball from the maze etc. Reading more from the opencv docs was useful as the solution was actually very simple.

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.