TASK – 4

1. MISTAKES- WE DIDN'T FIXED IR IN THE MID OF THE ROBOT , THAT'S WHY ITS TAKE MORE TIME TO TURN , AND IT HAVE TO GO FORWARD .
2. **METALLIC PEBBLE –** Make metallic peblle , total = 5*3=15 . diameter=2.5cm weight=4 to 5g
3. **Crow robot ID-** crow robot ID should be 10;
4. **BUZZER-** After all pebbles have been deposited, robot has to sound buzzer for 5 seconds continuous until the end of the run.
5. **AR PART**

   Teams have to write a python script which:

   1. Is run at the start of the run.
   2. Identifies ID and position Of AR_Objects placed in the Arena.
   3. Projects 3D models of pebbles on the Pebble AR_Objects.
   4. Projects 3D models of Water Pitcher/Crow on Water Pitcher AR_Objects/Crow.
   5. 3D Models are animated to depict pickup and drop of magnetic pebbles in the arena

6.

Task Flow

The task flow for the theme is as follows:

1. Python script (written by teams) is started. The image of arena is projected on screen with 3D models being projected on top of AR_Objects.
2. Robot is placed (switched OFF) on either START-1 or START-2 according to configuration.
3. Once team is ready, robot is switched ON.
4. Robot traverses to each of the Pebble AR_Object locations and picks up the magnetic pebble from under the AR_Object.
5. During pebble pickup, augmented reality part should show Pebble Pickup animation.
6. After each pebble pickup, robot traverses to Water Pitcher AR_Object to deposit the magnetic pebble in the deposition/pickup zone under the AR_Object.
7. During pebble drop, augmented reality part should show Pebble Drop animation.
8. After all magnetic pebbles have been deposited under the Water Pitcher AR_Object, robot must sound the buzzer for 5 seconds to end the task.

6. **FINAL ARENA SETUP**

Final Arena Setup

A sample arena configuration is given in Table 1 and Table 2.

AR_Object Type	ArUco ID	Cell No.	Orientation Axis
Water Pitcher	0	5	2-2
Pebble	1	3	1-1
Pebble	2	11	3-3
Pebble	3	13	2-2

Table 1: AR_Object Sample Configuration

Robot- Crow	
Aruco ID	10
Position	START-2

Table 2: Robot Start Position

7. SAMPLE ARENA CONFIG

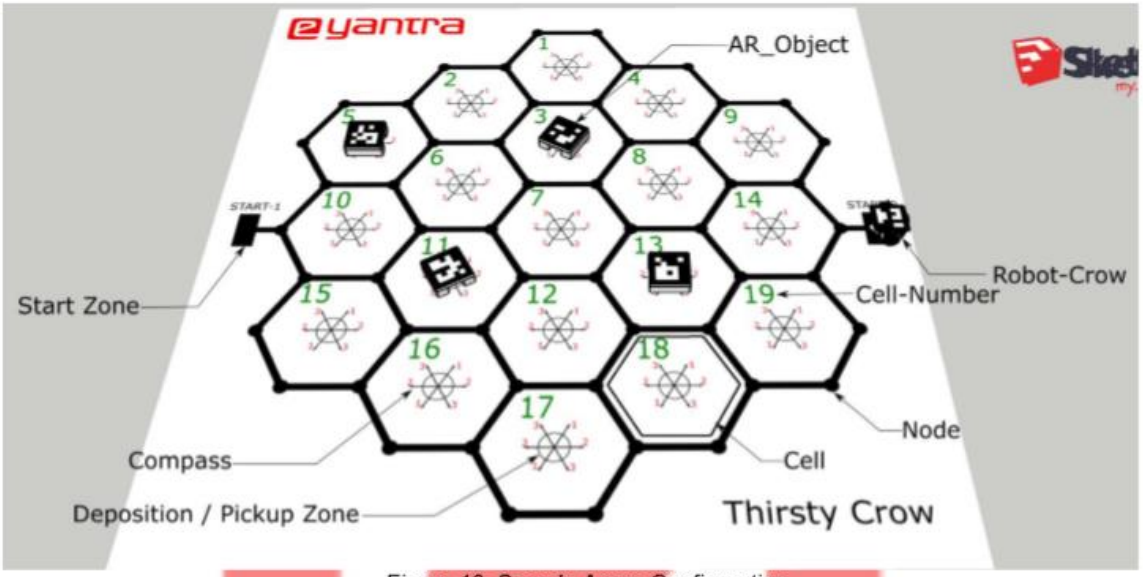

Figure 10: Sample Arena Configuration

8. INPUT OF ARENA CONFIGURATION IN THE PYTHON SCRIPT

The Arena Configuration is given in form of a dictionary as input to the python script written by the teams as given in Table 3.

CODE

`arena_config = {0: ("Water Pitcher", 5, "2-2"), 1:("Pebble", 3, "1-1"), 2:("Pebble", 11, "3-3"), 13:("Pebble", 13, "2-2")}`
 	
`Robot_start = "START-1"`

Table 3: Code to initialise configuration in python script

9. REPOSITIONING

8. Re-Position Rules

- While traversing the Arena, if the robot strays off the black line, a member of the e-Yantra team will place the robot on the previous node (node already traversed by the robot) in such a way that both the wheels of robot are parallel to the line and the white line sensors are above the Node. This is termed as **Re-position**.
- During Re-position, the timer will not stop and robot will not be switched off.
- Teams are allowed a maximum of 2 Re-positions. The run ends and timer is stopped if the robot requires a Re-position after 2 Re-positions have already been exhausted.
- 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.

Thirsty Crow - Revised Scoring Criteria

Dear Participants

We hope you guys are working hard for the Theme Implementation.

In the last week, we have reviewed the Progress Task submitted by all the teams and we have given helpful suggestions to the teams who submitted the video. Here we would like to discuss some changes made in the Final Scoring Criteria which has been given in the Rulebook.

Total Score = (300-T) + CPP*100 + CPD*100 + CARP*50 + CARA*100 - P*30 + DB + B
T - Time (in seconds)
CPP - Correct Pebble Pickup
CPD - Correct Pebble Drop
CARP - Correct AR Projection
CARA - Correct AR Animation
P - Penalty
DB - Design Bonus
B - Bonus

The changes made to the scoring criteria are as follows:

- **Correct AR Projection** - During the evaluation of Progress Task, we realized there is ambiguity in what is considered as Correct AR Projection. Henceforth, Correct AR Projection (CARP) is defined as projecting the correct Blender model (Pebble should be projected for Pebble AR_Object and Water Pitcher should be projected for Water Pitcher AR_Object) with **properly applied texture.**

  Examples are given as follows:

10.

BEST OF PIAZZA

1. IF ENCODER NOT WORK

If your robot is not stopping when the desired pulse count is reached, there must be something wrong with the code or the hardware connections

1) Check if you are supplying 5V and Gnd correctly to the motors. The internal position encoder circuitry requires those.
2) Make sure you are connecting the A and B ports to the correct interrupt pins
3) Double check all the wire connections. 60-70% of cases its because the faulty jumper wires or badly done circuit connections that problems develop in the circuit.

On the code side,
1) Make sure the interrupts are initialised properly
2) Use the volatile keyword for the pulse count variable that you are incrementing in ISR

If it still doesnt work, please share your code, we will have to go through it. (**BUT ONLY THE INTERRUPT CODE**).

2. FOR CAMERA MATRIX AND GOOD PROJECTION

Try the camera calibration for a larger number of images and from a greater number of angles.

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_pose/py_pose.html

This tutorial might be helpful if you want to undistort the camera frame

As far as i can see, there is a minor offset where your models are being projected. That also might be if the blender models you have created are not placed at origin in the blender scene.

3.