Introduction to ROS

Project Report

Yang Wang
yang.wang@estudiantat.upc.edu
Raphaël Atlan
raphael.atlan@estudiantat.upc.edu

June 2021

1 Introduction

The aim of this final project is to put in practice all the ROS tools that have been seen so far, in order to make the ur3 robots move the chess pieces:

A chess match will take place between a human and a robot. The robot will be responsible of moving the pieces according to the chess motions given by a user (or optionally by a chess engine).

There are 8 aruco markers on the chessboard, with labels 100 to 107, that will be used to calibrate the camera pose. The pieces have also aruco markers with the numbers to differenciate them. The camera can detect each marker and give it's number and the corresponding pose.

The development of the project can be split in the following three modules:

2 Sensing module

It should be the responsible of sensing the state of the environment. It should be able to:
• Detect and set the poses of the chess pieces.
• Give the coordinate of a given chess pieces.
• Give the chessboard cell where a given chess piece is located.

3 Planning module

It should be responsible of planning the sequence of motions to execute a chess action. The following services and function could be provided to:
• Compute the robot pose to pick/place a piece in any given cell.
• Plan a collision-free rectilinear motion.
• Plan the motions for a pick action.
• Plan the motions for a place action.
• Parse the output of the task planner into the corresponding action.

4 Action manager module

It should be the responsible of launching the execution in the Gazebo simulation and in the real environment. It should provide functions and services to:
• Open/close the gripper.
• Execute a robot motion.
• Receive the chess action query to be executed.

• Manage the whole process.

5 Test goals

The following tests will be used to evaluate the performance of the implemented system:
• Locate a given piece (returning the cell code where it is placed).
• Plan the motion of the robot (without holding any piece) from the home configuration to a pre-grasp configuration on a given cell (specified by the chess coding, e.g. B4) and visualize the path in rviz.
• Plan the motion of the robot (holding a given piece) from the home configuration to a pre-place configuration on a given cell.
• From a given pre-grasp configuration (with the gripper closed without holding any piece) plan a pick motion of a particular piece.
• From a given pre-place configuration (with the gripper closed holding a particular piece) plan a place motion to place the piece on the cell.
• Extend the third test to the case where the piece has to be placed at the safe-region used for the castle operation, or at the region where to the killed pieces are thrown.
• Plan the motions to move a piece.
• Plan the motions to kill a piece.
• Plan the motions for a castle operation.
• Execute any of the previous plans in Gazebo and in the real robot.

6 Development

6.1 Manager node

The system is handled by a manager node. It has the following functions:
• Receive a command from a chess agent.
• Call the sensor to obtain the sensing messages.
• Transform the sensing messages to map between the chess name and its location.
• Compute inverse kinematics for a given pose.
• Check a given pose is collision free or not.
• Call the controller to execute a collision free trajectory to a given pose.
• Determine to perform a simple movement or a kill action.
• Manage the whole process.

The manager node is connected with different modules, the connection can be illustrated in the Fig. 1.The manager node connect each module by services that are provided either by a designed function node or a provided services in the chesslab setup environment.

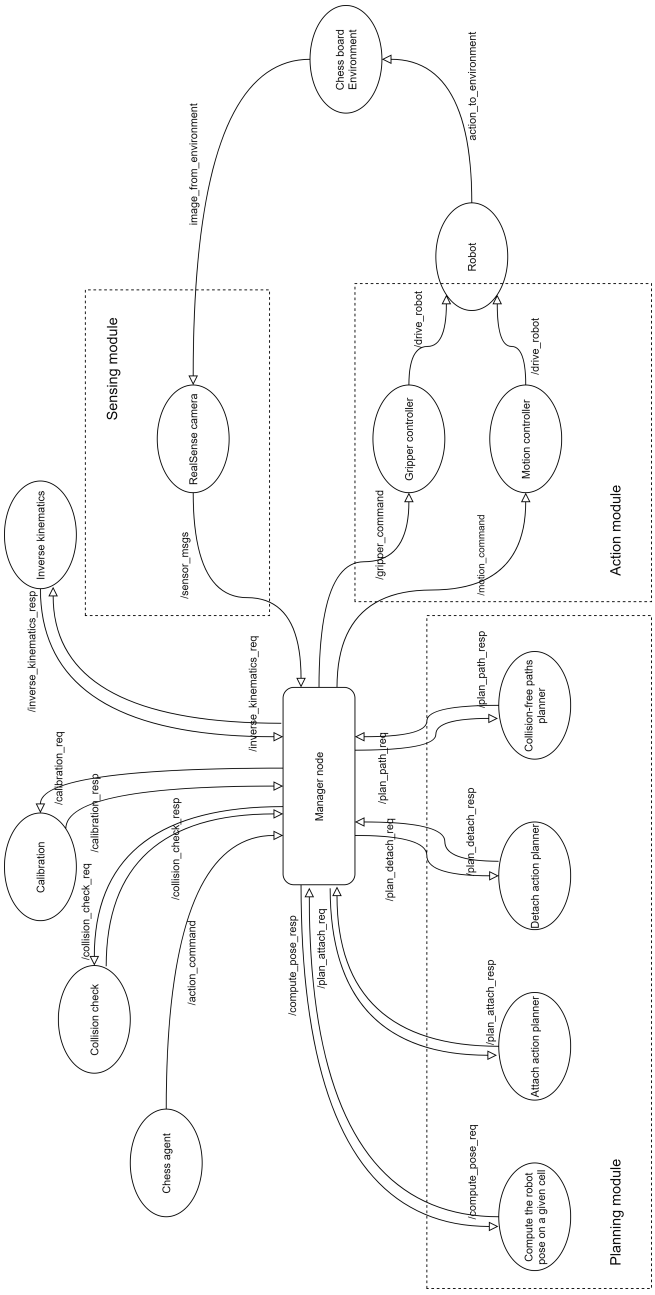

Figure 1: Communication between each node

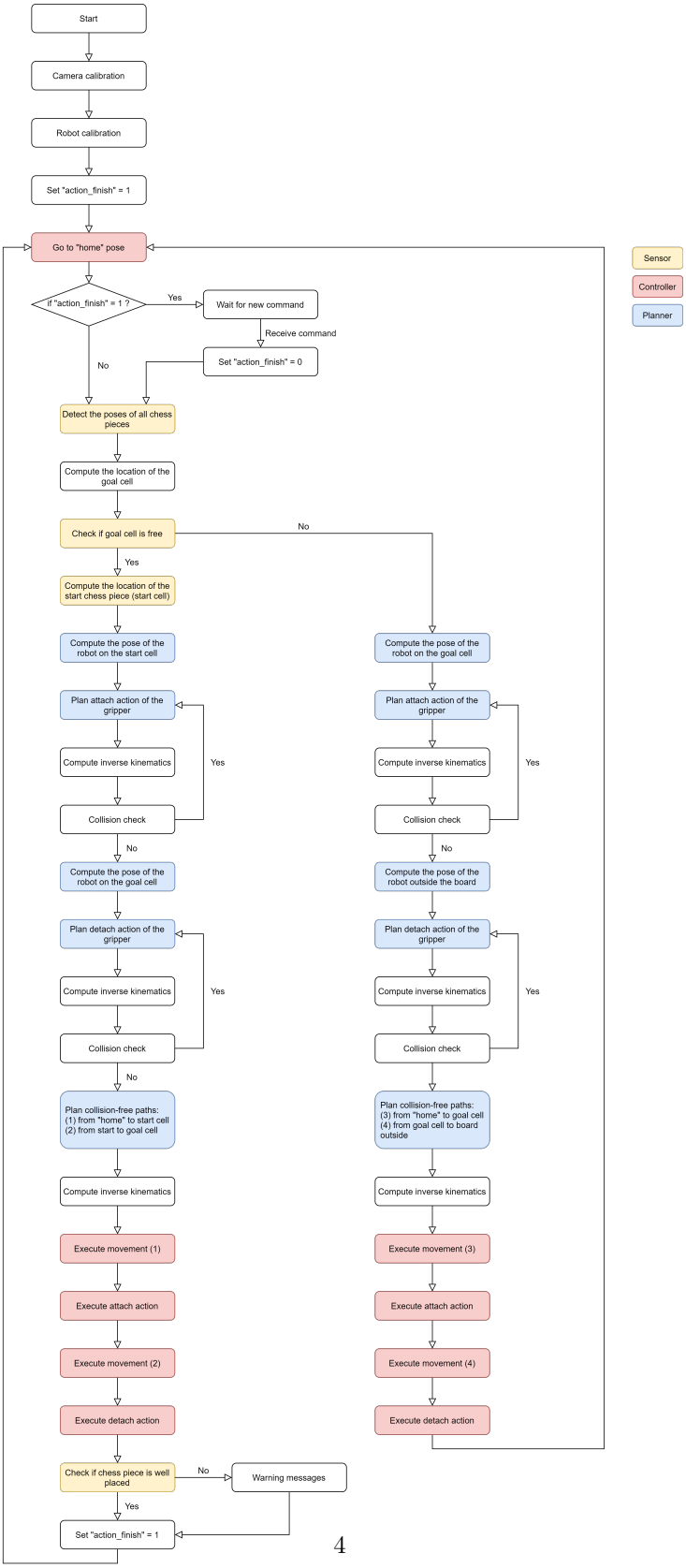

Figure 2: Communication between each node

The entire working process can be shown in Fig. 2. The main procedure has three phases: Sensing $->$ Planning $->$ Control. The main features of each phase are described below.

6.2 Sensing

The aruco broadcast node sends the aruco markers information in a array. Each element contains the marker ID and its pose. The broadcasted poses are originally related to the camera frame. To obtain the valid information we us a tf2::Transform to do a coordinate transformation for each of the marker. The transform is done from camera frame to the world frame.

After receiving the poses of all the marker, we need to build a map between the marker and the chess pieces and the location in terms of the chessboard cell. Since each chess has an unique id that is known, it is simple to link between chess name and its cartesian coordinate. To obtain the cell location of a chess, we first build a map that relate the cartesian coordinate and the cell code, afterward, using this map, we can transform the catrisian coordinate of the chess pieces to their cell location.

6.3 Planning

The planning module is in charge of giving a plan from a start pose to a given pose. The planned paths are rectilinear trajectories, which is determined in the joint configuration space. The planned paths includes the following types:
• Path from a initial pose to the pre-defined home pose.
• Path from the home pose to a given pre-pick pose.
• Path from a pre-pick pose to the pose that is ready to pick a chess piece.
• Path from a pre-pick pose to a pre-place pose.
• Path from a pre-place pose to the pose that is ready to place a chess piece.
All the proposed path from planner will be sent to the manager node and check for collision. If a given trajectory is not collision free, the planner will give another possible path. If all possible path are not collision free, the system will ask for a new command that is move to other poses.

6.4 Controller

The controller consist of two parts: the ur3 arm robot controller and the gripper controller. In this project these two parts are used separately.

For ur3 arm robot, we use action client to execute a planned trajectory. These trajectory is defined in the robot joint configuration space. The gripper has two parameters to be tuned, the gripper open angle and the effort. They are set dedicatedly to perform a proper grasping operation.

7 Execution procedure

The complete execution procedure is below:
1. Initialize the system, move the arm to a given home pose.

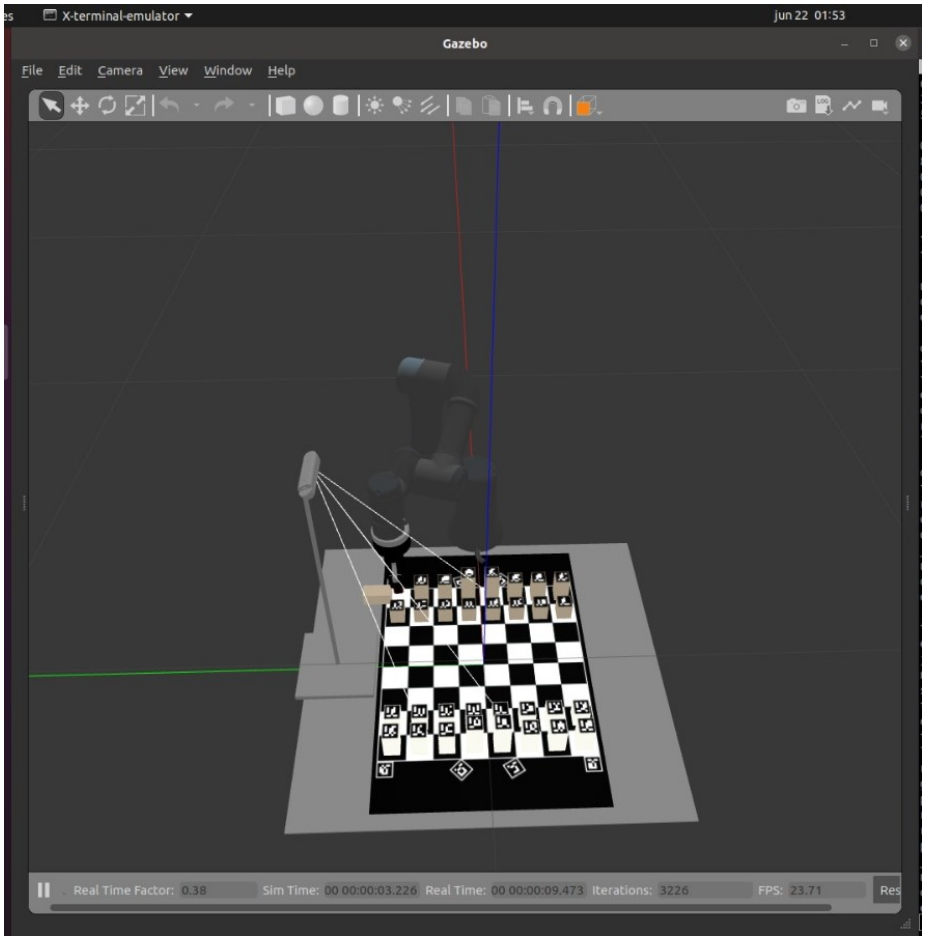

Figure 3: Robot in initial pose

2. Sensing the chessboard environment, compute the mapping between the aruco markers and the chess name and the cell location of each chess. Once it is done, the initialization is finished.

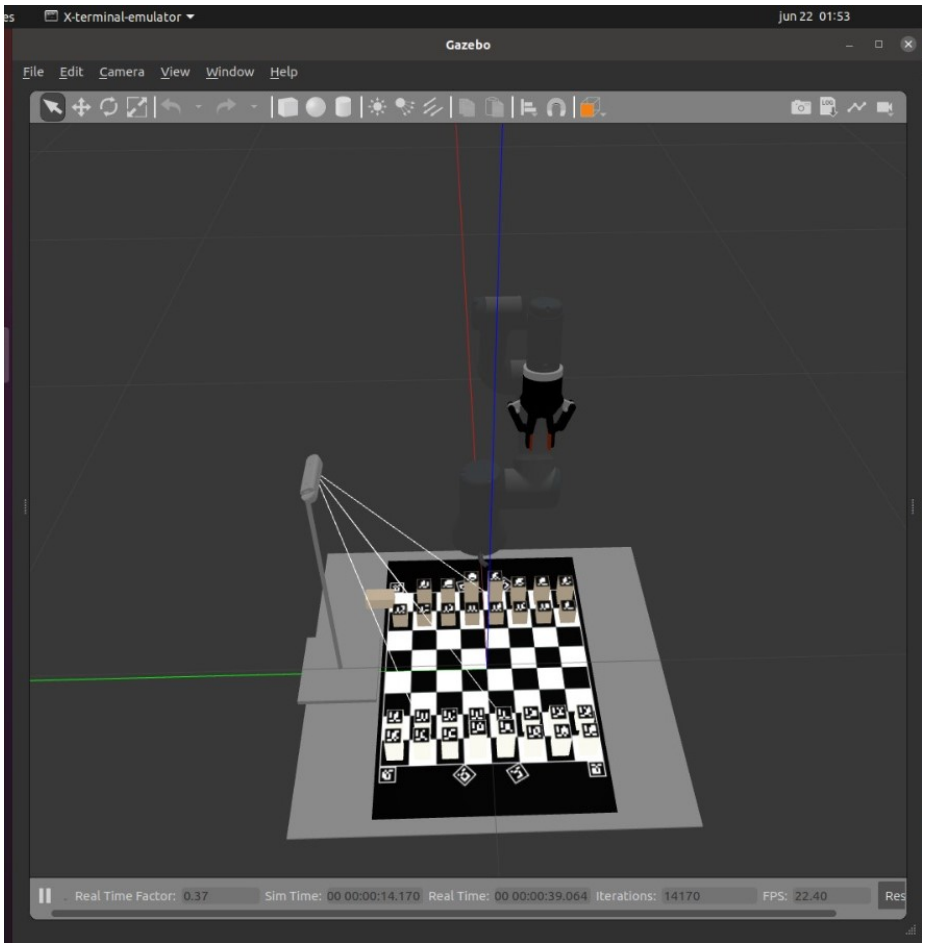

Figure 4: Robot finish initialization

3. Wait for chess agent to send a command. This command has the following format: ["chess name","goal cell"].

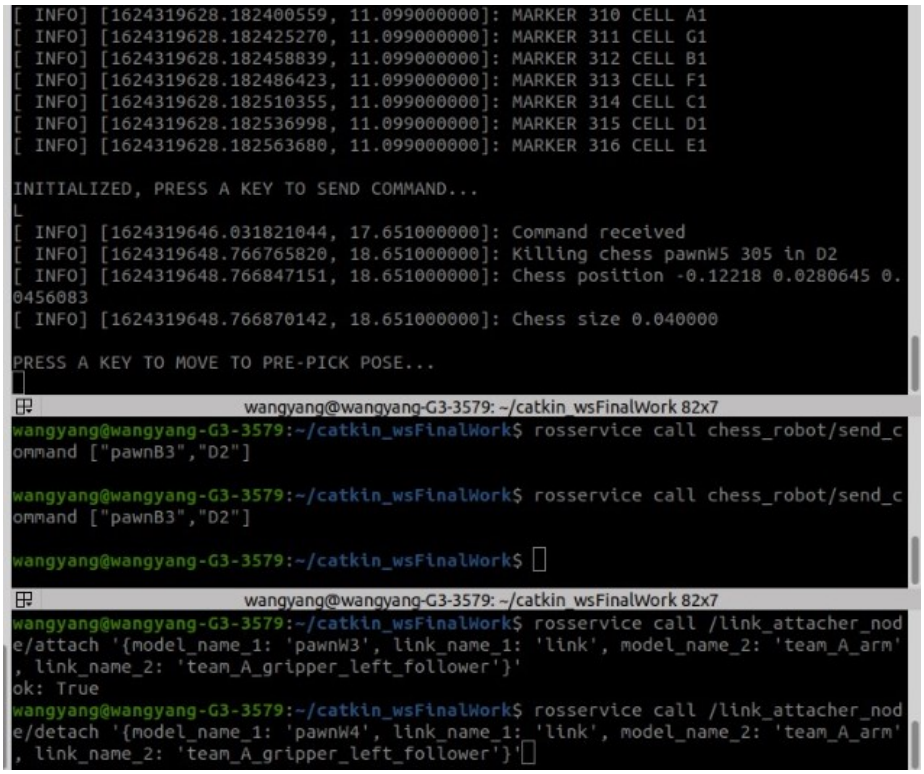

Figure 5: Robot receive a command to move black pawn 4 to cell D2

4. Manager node determines that if the goal cell is free. In case it is not free, this step is treated as a "kill", otherwise a simple movement. The following steps correspond a "kill" operation. Simple movement is simplified version of it.
5. Manager determines the pose of the piece that should be killed, and compute a pose that is higher such that the arm can ready to pick the chess piece without collision.
6. Manager node calls the inverse kinematics service to compute the corresponding joint configurations, and for a computed configuration, calls the collision check service to determine that this configuration is collision free.
7. Manager node sends a collision free trajectory to the controller.

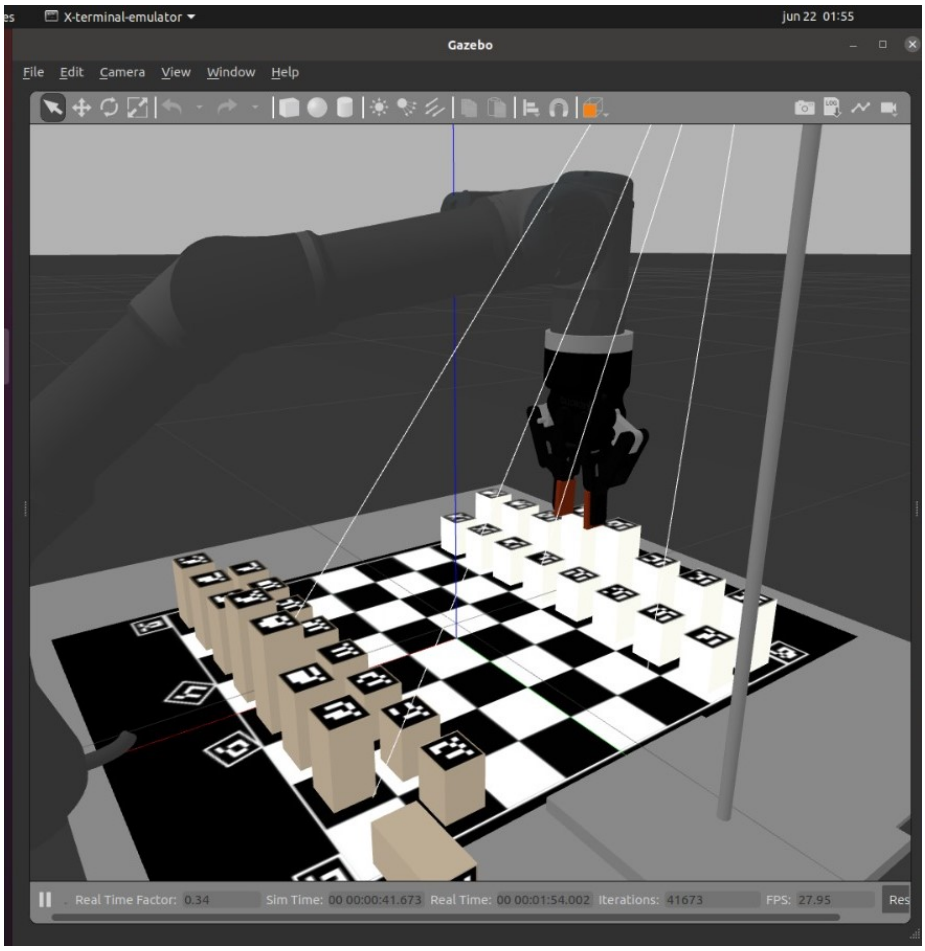

Figure 6: Robot is in a pre-pick pose

8. Controller first executes the robot arm trajectory, once the arm is moved to a pose that is ready to pick a chess, it will close the gripper to grasp the chess piece.

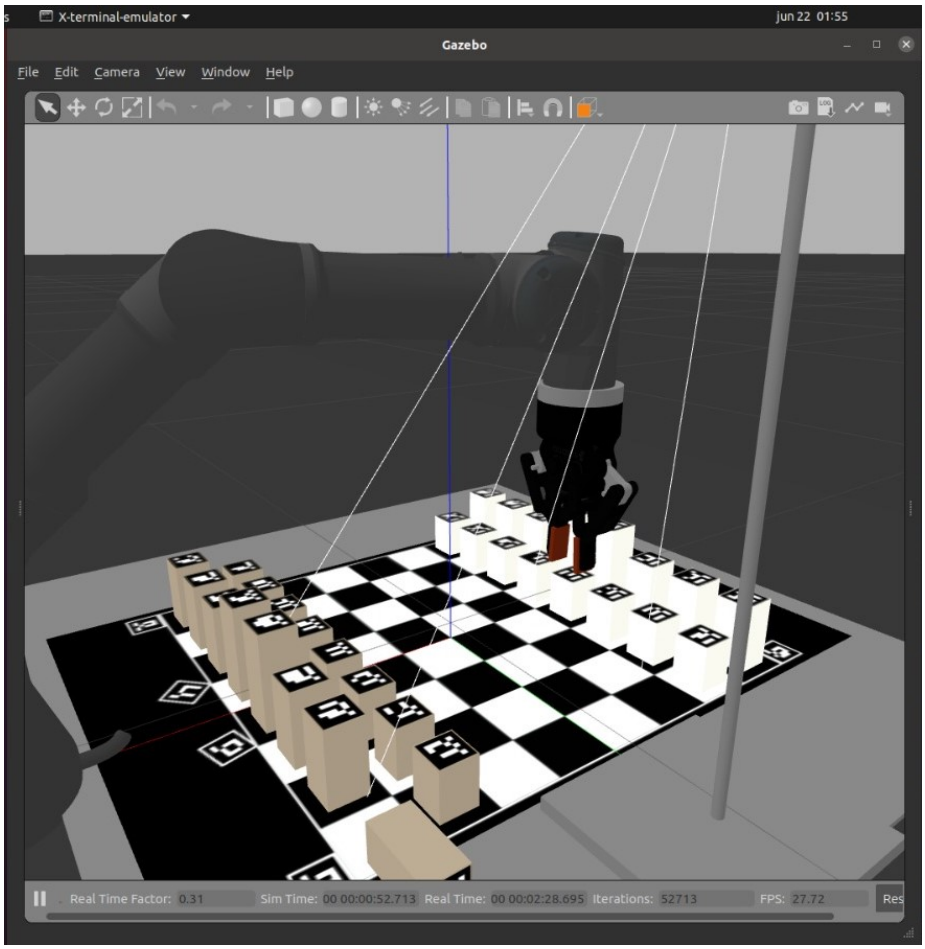

Figure 7: Robot is picking a chess piece

9. Manager node calls the planning module to give a trajectory to the safe-zone, that is used to place the killed chess piece. This trajectory is checked to be collision free.
10. Controller execute a collision free motion to place the killed chess piece to the safe-zone.

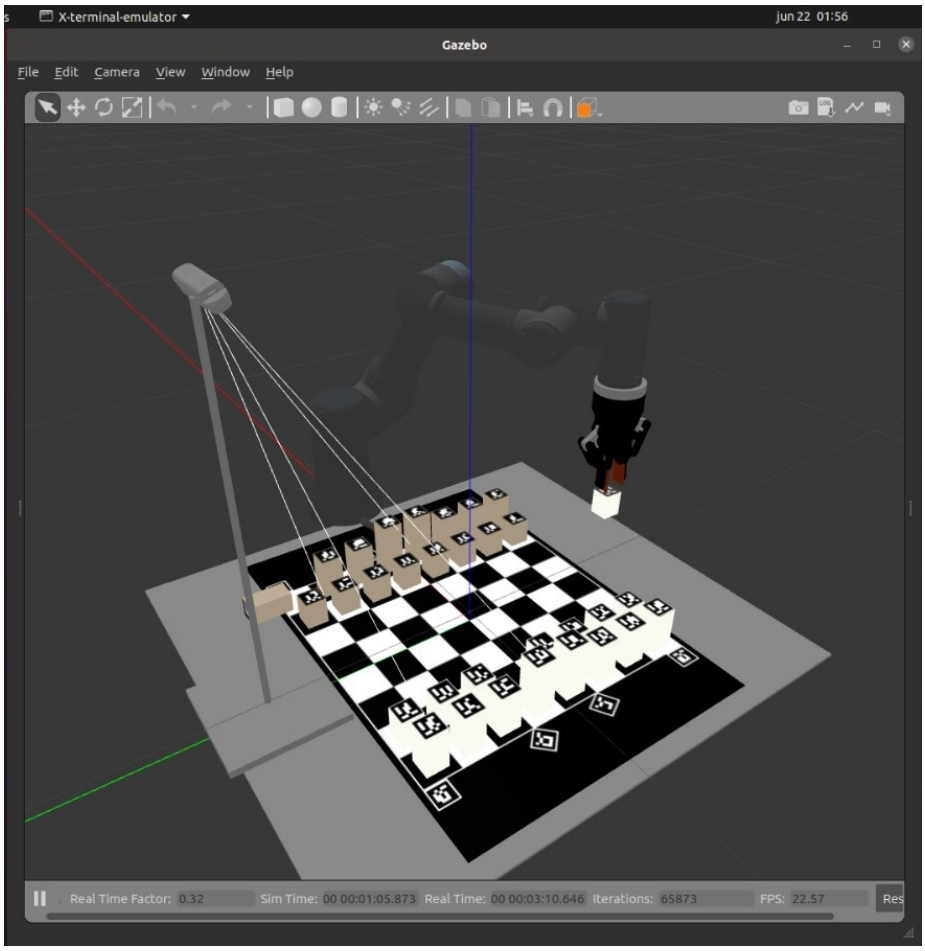

Figure 8: Robot moving a chess piece to the safe-zone

11. Robot go to home pose, and calls the sensor module to re-scan the environment.

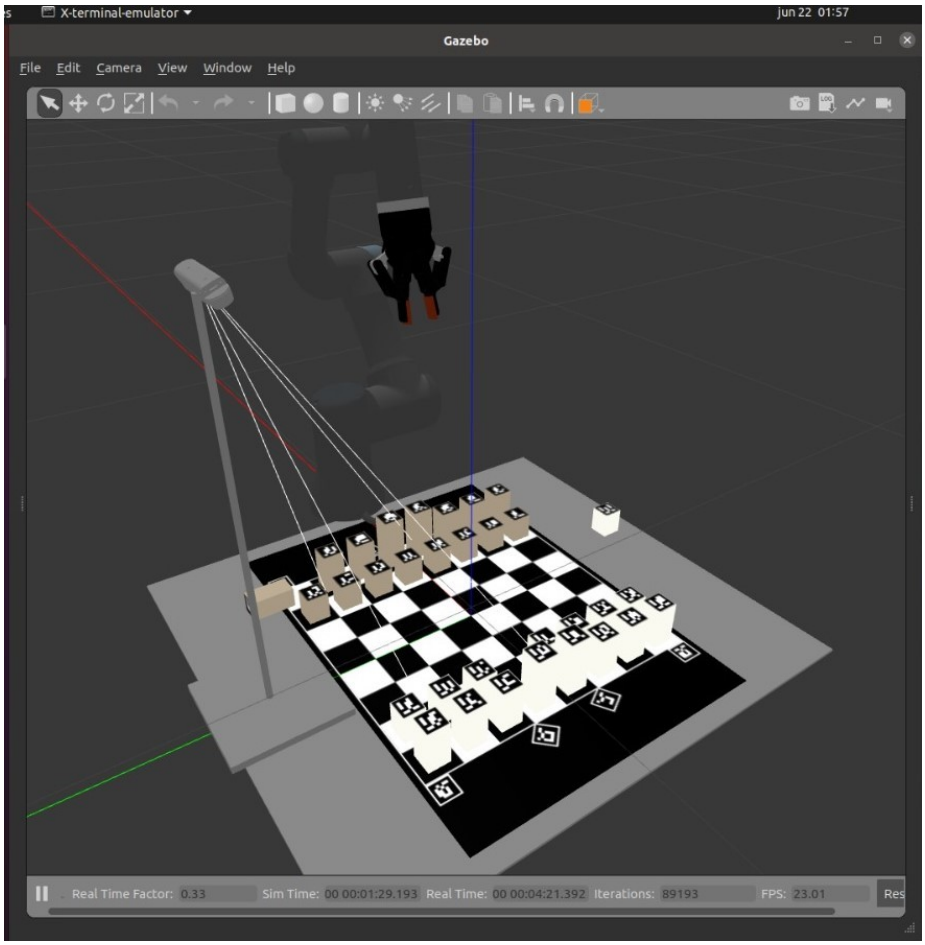

Figure 9: Robot moves to the home pose and scan the environment again

12. Manager node perform a simple movement operation. Following the similar steps as in killing operation, put the target chess to the goal cell.

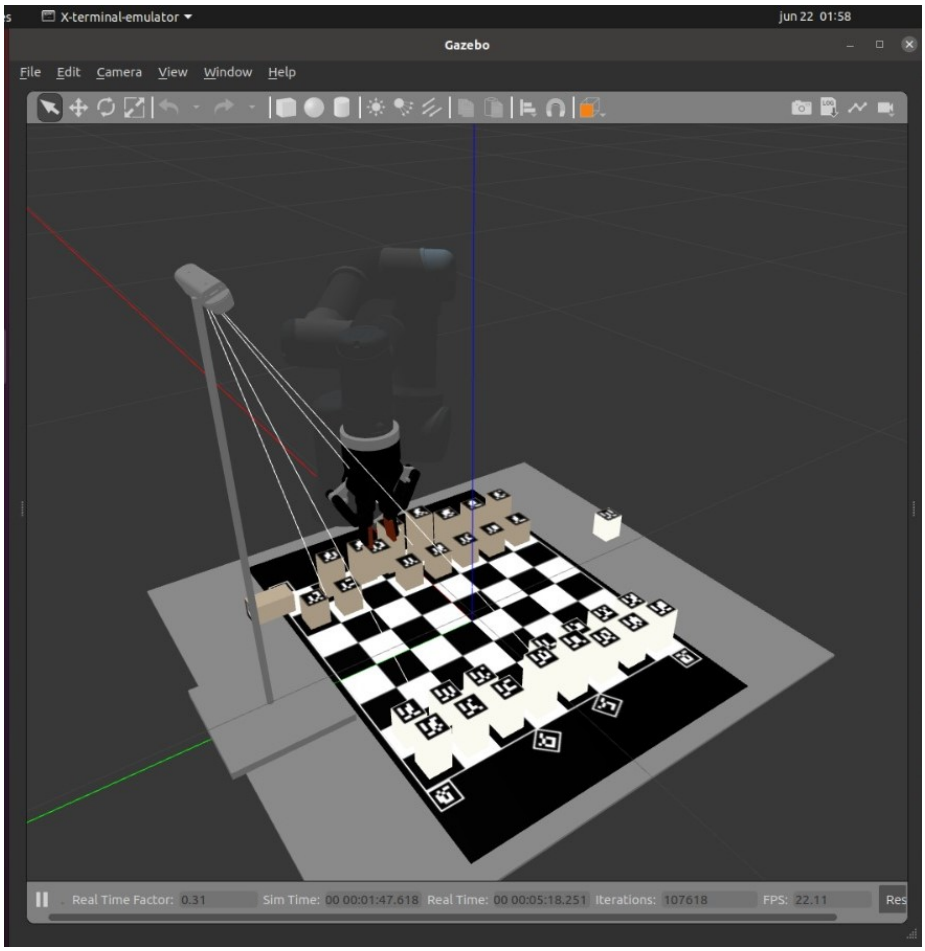

Figure 10: Robot pick a chess piece to place to the goal cell

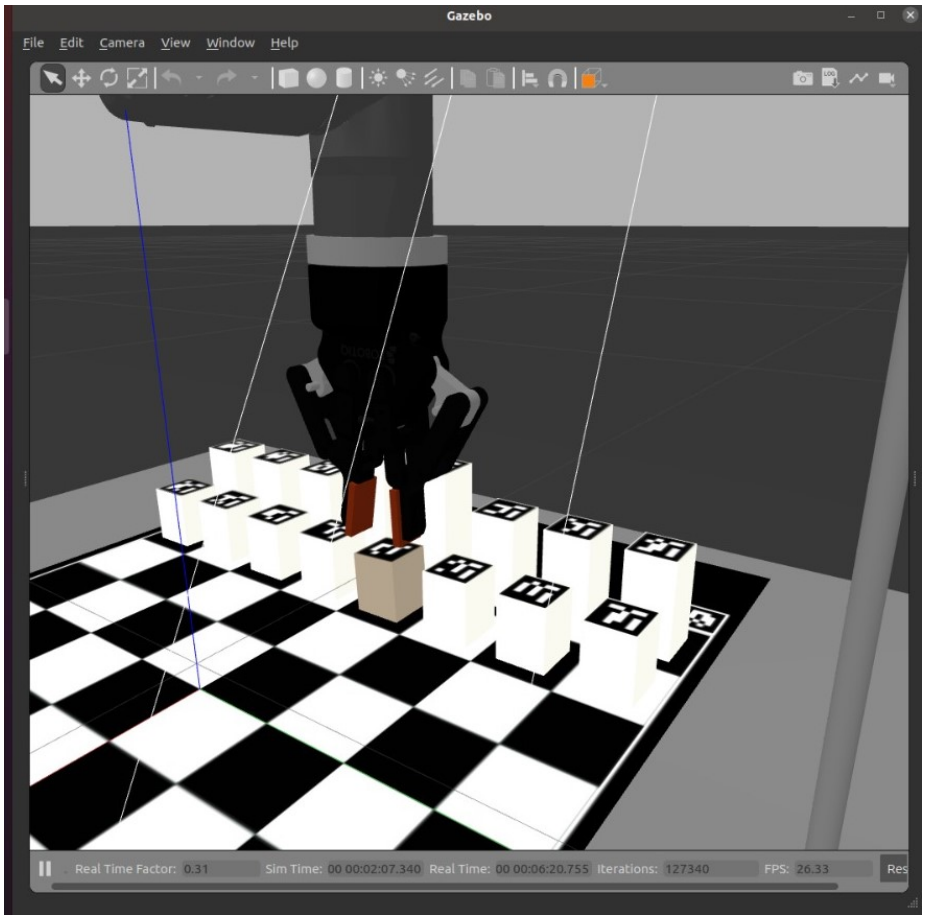

Figure 11: Robot placing a chess piece to the goal cell

13. One round is finished, the robot will go back to home pose and scan the environment again and the system will wait for the next command.

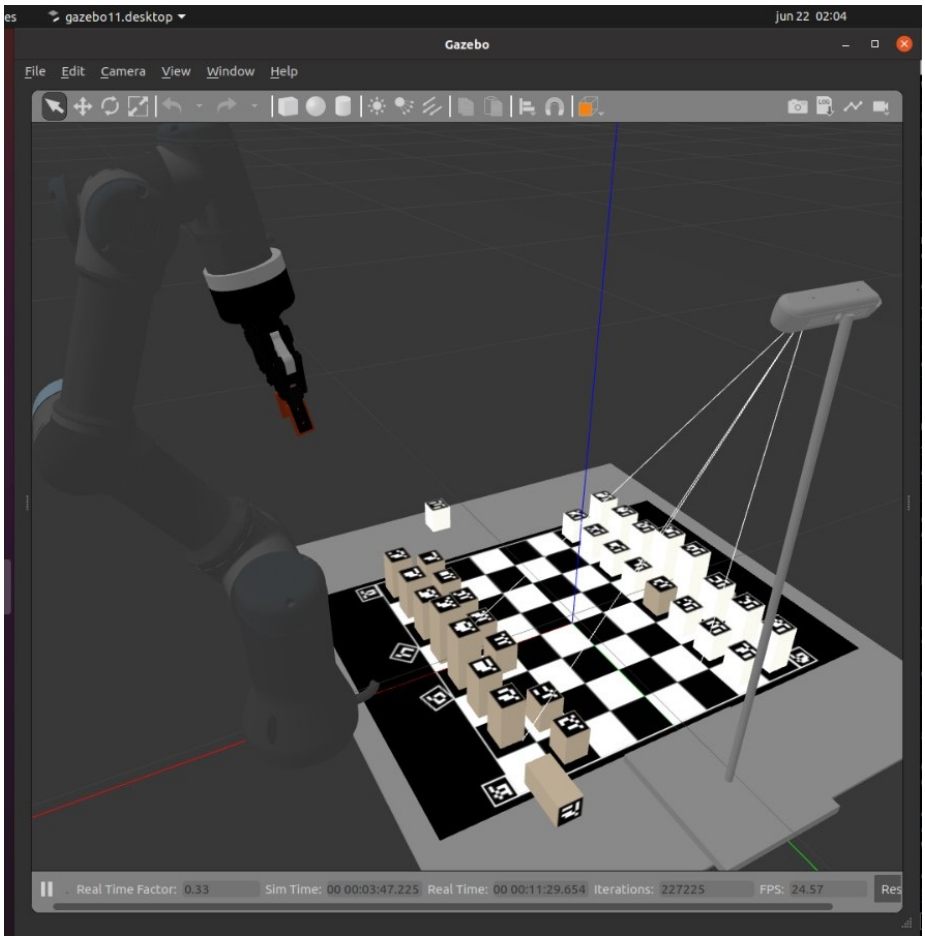

Figure 12: One round finish wait for the next command

Until this, the main working process of the developed system is illustrated.