Joe Garfield,
Team N. 6. Manohar Darapurddi,
Jianwei Zuhang,
Pedro A. Diaz

SOFTWARE DESIGN.

PROJECT 3.

DELIBERATION AND ACTING.

Following some of the Murphys' guidelines in the design of a behavioral System we work with the hardware sub-team in:

- Reading and understanding the specification of project n. 3 which are: to design, build, program, and demonstrate an autonomous robotic system that carries out the task to *efficiently* find and move a set of target objects in the environment to one or more locations within the environment and, if possible, to find and disable a moving dynamic object.
- Describe the task and the environment which are: Details of the task are not going to be enumerated here for briefness.
- Describe the robot: which is described in the hardware report.
- Describe how the robot should act in response to its environment: In this part we worked first in the **primitive behaviors**:

  move_t(): move_to_target(), the robot moves from its actual position (r_x, r_y) to the nearest orange cube.

  move_d(): move_to_destination(), the robot moves from its actual position (r_x, r_y), with an orange cube, to the nearest square destination.

  cam(): look_for_target(), the robot looks for orange cubes, using the camera.

  tape(): look_for_destination(), the robot looks for square destinations, using the IR-sensors.

  look_for_DrH_robot(), the robot looks for Dr. Hougen's robot using the camera and

  signal_done() when the robot puts an orange cube in the destination it beeps.

Second, we worked in the **deliberative part** and we design the following functions:

The mission_planner which correspond to our **main()**, it receives the coordinates of the targets, destinations and the initial robot's position, it initialize the **distance_t()** process which call the move_t(); the **color()** process, which look_for_target() and directs the robot to go to the nearest destination with the function **distance_d()**. This process looks_for_DrH_robot() too, using as a perceptual schema the blue color.

Joe Garfield,
Team N. 6. Manohar Darapurddi,
Jianwei Zuhang,
Pedro A. Diaz

The **navigator** was implemented implicit in the move_t() and move_d() functions. The robot is moving in straight lines following the “x” and “y” directions so that, each path consist of a triangle path.

The **pilot** correspond with the functions: move_t() and move_d() which accomplish the necessary actions to go from one position to another, for example, turn(90); forward(100).

The **cartographer**, which has track of the orange cubes, square destinations and the actual position of the robot. This function is distributed between various processes: the **color(),** because if a cube is moved from its position to the destination, the coordinates of the cube are set to (-1,-1), and if an unknown cube is encountered, the world model is actualized with the coordinates of the new cube which correspond to the current position of the robot; by the other hand the function move_t() actualizes the WM with the current position of the robot, this is done at the end of each segment.

The low_level controller correspond with the functions: forward(x), which move the motors forward; turn(y), which turns the robot y = 90 or y = 180 grades;

The **sensing** part correspond with: **encoder_dist(x,*y)** which read the encoders to have track of the distance walked and actualizes the WM; **tape(),** which uses the IR sensors to look for the black tape destinations; the **cam(),** which is looking for the orange and blue colors.

In one word, we follow the Managerial Architecture.