BLOCK : TASK MANAGER

ESCAPE ROOM CHALLENGE

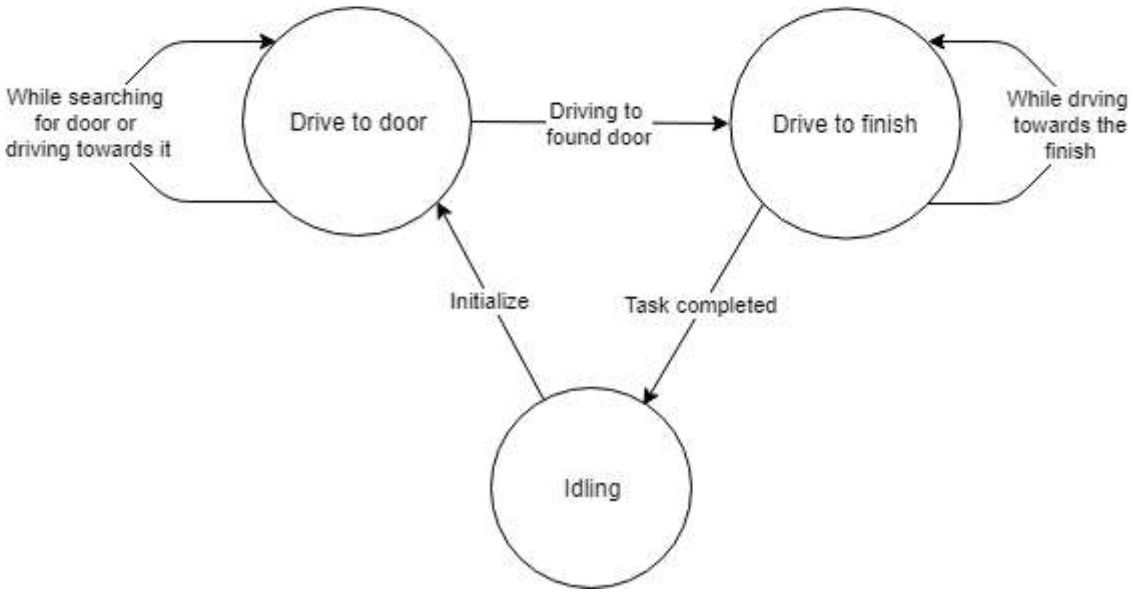
The task manager functions as a finite state machine which switches between different tasks/states. It focuses mainly on the behavior of the whole program rather than the execution. It determines the next operation phase and case based on the current phase, current case, current block statuses and counters in which the corresponding block modes of the perceptor, path planner and drive controller are set for the upcoming execution. It communicates with the other blocks via the World model. Since the "Escape room challenge" and the "Hospital competition" require a complete different approach in terms of cooperation between the blocks, the task planner is completely rewritten for both challenges.

INITIALIZATION:

The path planner is given a command “Drive_to_door” while the drive controller and the preceptor are given a command “Execute” as a part of the initialization process.

EXECUTION:

The high-level tasks “Drive_to_door”, “Drive_to_exit”, “Execute”, “Idle” and “Disable” were given to appropriate blocks as shown in figure 9. If the status of the Path planner is driving to a possible door or is searching for a door, the task manager sets the mode of path planner to drive to door. If the status of the path planner is driving to a found door and the Drive controller is done driving, then the path planner is set to drive to exit. If the status of the Path planner is driving to a possible exit or is searching for a exit or if the drive controller is busy driving, the task manager sets the mode of all the blocks to Idle.



INPUTS	Drive Controller status, Perceptor status, Path Planner status.
OUTPUTS	Drive Controller modes, Perceptor modes, Path Planner modes.

TASK MANAGER FUNCTIONALITY :

FUNCTION	**FUNCTION DESCRIPTION**
Init()	To initialize the Drive controller to 'Execute', Path planner to 'Drive_to_door' and Perceptor to 'Execute'
Execute()	1) If (PathPlanner_Drving_to_PossibleDoor) or (PathPlanner_Searching_for_door) then, the blocks are set to (PathPlanner_Drive_to_Door) and (DriveController_Execute) and (Perceptor_Execute). 2) If (PathPlanner_Driving_to_FoundDoor) and (Drivecontroller_Done) then, the blocks are set to (PathPlanner_Drive_to_Finish) and (DriveController_Execute) and (Perceptor_Execute). 3) If ((PathPlanner_Driving_to_Finish) and (DriveController_Busy)) or (PathPlanner_Searching_finish) then the blocks are set to (PathPlanner_Drive_to_Finish) and (DriveController_Execute) and (Perceptor_Execute). 4) If (PathPlanner_Driving_to_Finish) and (DriveController_Done) then the blocks are set to (PathPlanner_Idle) and (DriveController_Disable) and (Perceptor_Disable).

HOSPITAL ROOM CHALLENGE

The task manager is completely revamped for the hospital challenge. It functions as a state machine and handles the behavior of the program in a well structured manner considering various fall back scenarios which can be edited when ever needed. The list of cabinets to visit are initially read by the task manager and sent to the world model. It works primarily on setting appropriate block modes, setting counter variables and changing from a particular phase and case of the program to another in order to perform a required task based on the block statuses it receives. The various phases and cases are discussed below :

PHASES AND CASES :

1. **INITIALIZE :**

The software always starts in this phase. During this phase all the inputs and outputs of the robot are initialised and checked. Also all the required variables in the software are set to their correct values. At the end of the initialisation phase the software is set and ready to perform the desired tasks, the software switches to the fitting phase.

- Case 1 :

- Init blockmodes of init values for all blocks
- Set blockstatusses to init values for all blocks
- Read and store high level tasks
- Read taskplanner behavior from file and store

2. FITTING

During the fitting phase PICO tries to determine its initial position relative to the given map. It determines the location with the help of the laser range finder data and tries to fit the environment around the robot to the given map. In the case that the obtained laser data is insufficient to get a good, unique fit, it first starts to rotate the robot. If after the rotation still no unique fit is obtained, the robot will try to drive towards a different location and rotates again at that location. The full details on how the fitting algorithm works are described in the perceptor section of this wiki. As soon as there is an unique and good fit, the location of PICO is known and the software switched to the relocate phase.

- Case 1 :

- Fit map at current position.

- Case 2 :

- Set desired position of robot to rotate robot for fixed (configurable) angle.

- Case 3 :

- Drive to location

- Case 4 :

- Set desired position to middle of the room (optional, lower priority)

3. RELOCATE

During the relocate phase the goal is to move the PICO robot to the desired cabinet. To do this, a path is calculated from the current location towards the desired cabinet in the path planner. The drive controller follows this path and avoids obstacles on its way. When it is found that the path as a whole is blocked, a new path is calculated around the blockage. As soon as the PICO robot has arrived at the desired cabinet the software switched to the cabinet action phase.

- Case 1 :

- Calculate path to goal

- Case 2 :

- Proceed along path, set new desired position

- Case 3 :

- Drive to next node

- Case 4 :

- Brake link between nodes

4. ACTION

During the cabinet action phase the PICO robot executes the required actions at the cabinet. This includes saying 'I arrived at cabinet ...' and taking a snapshot of the current laser data to proof that the robot has arrived at the correct location. After performing the required actions the software determined if the PICO robot should visit another cabinet, and if so, it switched back to the relocate phase. If all the cabinets are visited the software is stopped and the challenge is successful completed.

- Case 1 :

- Store laser data

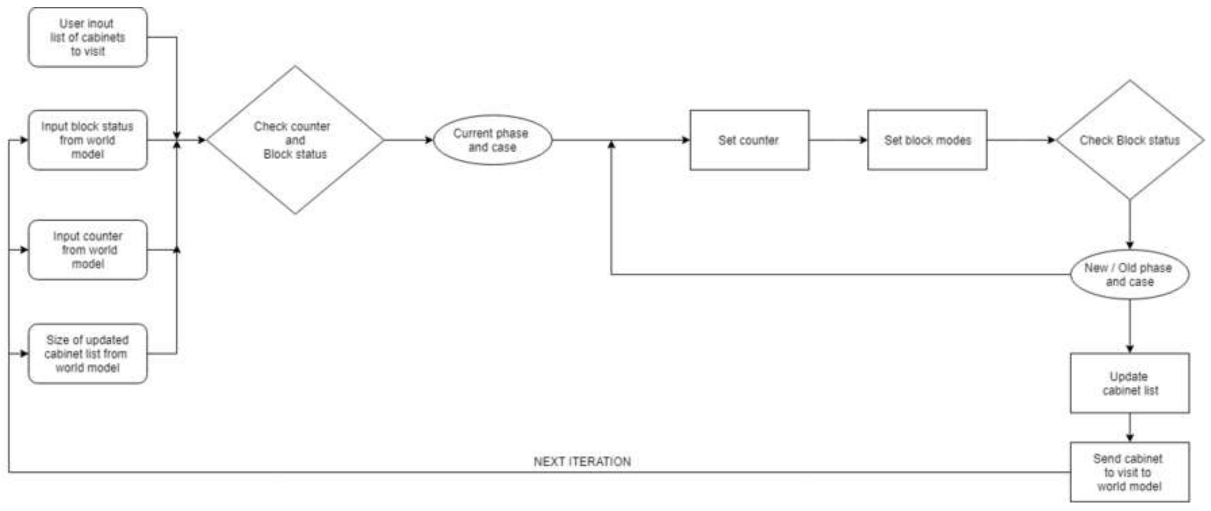
- Case 2 :

- Play sound

5. ERROR

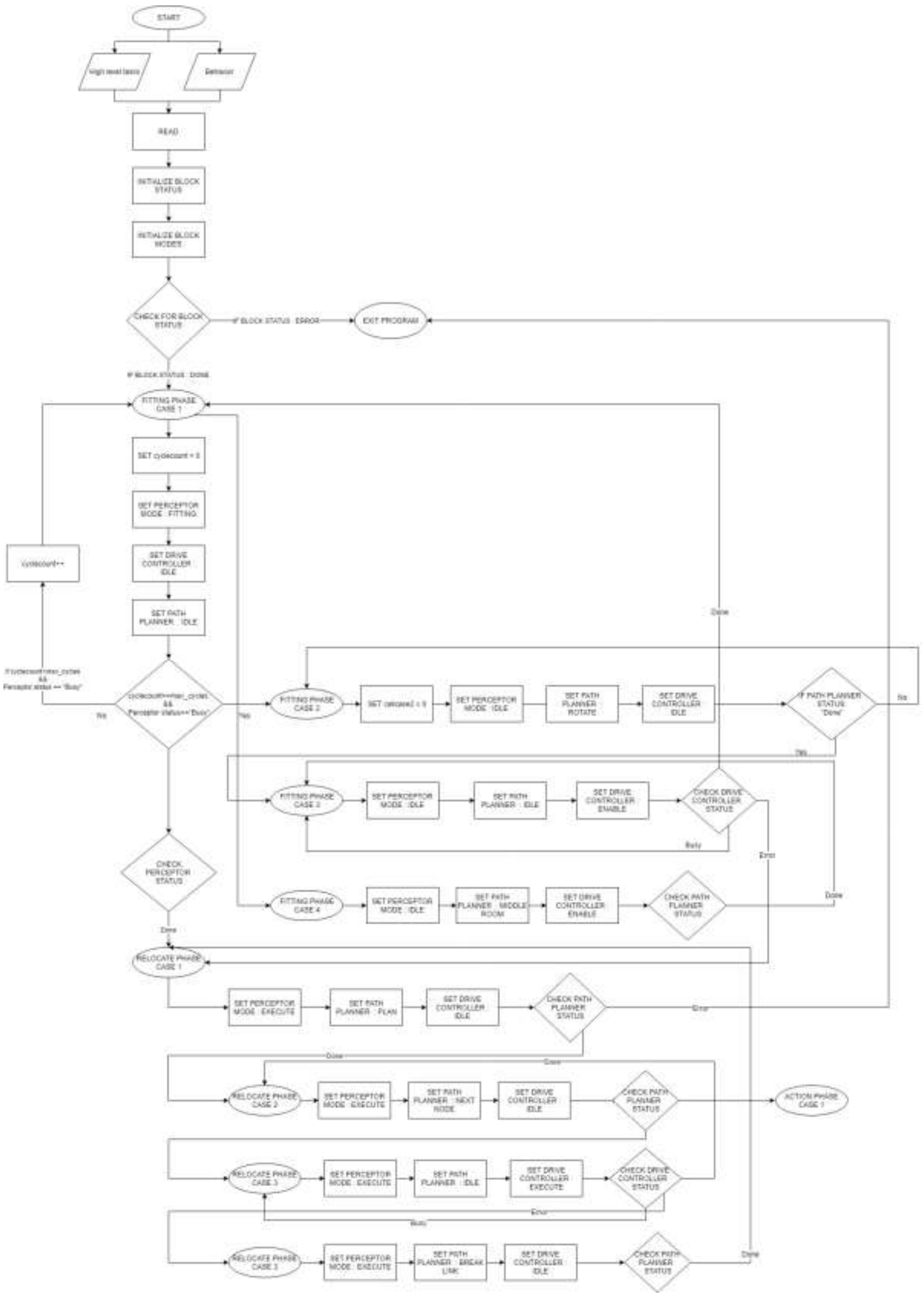
The error phase is different from the other phases in the sense that it is never the desired to end up in this phase. The only situation when the software switched to the error phase is when something is different from expected. This can for example happen when a required file is missing during the initialisation phase, the fitting phase has went through all the fallback mechanisms and still hasn't succeed in finding a unique fit. In all cases something unforeseen has happened and the software is out of options to recover itself. If that happens, the PICO robot is switched to a safe state e.g. all motors are disabled, and then all potential useful information for debugging is displayed to the operator. Finally the software is terminated and the possible cause can be found and fixed.

STATE MACHINE :


User inout list of cabinets to visit
Input block status from world model
Input counter from world model
Size of updated cabinet list from world model
Check counter and Block status
Current phase and case
Set counter
Set block modes
Check Block status
New / Old phase and case
Update cabinet list
Send cabinet to visit to world model
NEXT ITERATION

The flow chart below explains the setting of modes, counters and transition phases & cases based on status of the blocks received.

TASK MANAGER FLOW CHART :


START
High level tasks
Behavior
READ
INITIALIZE BLOCK STATUS
INITIALIZE BLOCK MODES
CHECK FOR BLOCK STATUS
IF BLOCK STATUS : ERROR
EXIT PROGRAM
IF BLOCK STATUS : DONE
FITTING PHASE CASE 1
SET cyclecount = 0
SET PERCEPTOR MODE : FITTING
SET DRIVE CONTROLLER : IDLE
SET PATH PLANNER : IDLE
cyclecount++
If cyclecount!=max_cycles && Perceptor status == "Busy"
cyclecount==max_cycles && Perceptor status=="Busy"
No
Yes
FITTING PHASE CASE 2
SET cyclecount2 = 0
SET PERCEPTOR MODE : IDLE
SET PATH PLANNER : ROTATE
SET DRIVE CONTROLLER : IDLE
IF PATH PLANNER STATUS "Done"
No
Yes
Done
FITTING PHASE CASE 3
SET PERCEPTOR MODE : IDLE
SET PATH PLANNER : IDLE
SET DRIVE CONTROLLER : ENABLE
CHECK DRIVE CONTROLLER STATUS
Busy
Error
FITTING PHASE CASE 4
SET PERCEPTOR MODE : IDLE
SET PATH PLANNER : MIDDLE ROOM
SET DRIVE CONTROLLER : ENABLE
CHECK PATH PLANNER STATUS
Done
CHECK PERCEPTOR STATUS
Done
RELOCATE PHASE CASE 1
SET PERCEPTOR MODE : EXECUTE
SET PATH PLANNER : PLAN
SET DRIVE CONTROLLER : IDLE
CHECK PATH PLANNER STATUS
Error
Done
Done
RELOCATE PHASE CASE 2
SET PERCEPTOR MODE : EXECUTE
SET PATH PLANNER : NEXT NODE
SET DRIVE CONTROLLER : IDLE
CHECK PATH PLANNER STATUS
ACTION PHASE CASE 1
RELOCATE PHASE CASE 3
SET PERCEPTOR MODE : EXECUTE
SET PATH PLANNER : IDLE
SET DRIVE CONTROLLER : EXECUTE
CHECK DRIVE CONTROLLER STATUS
Error
Busy
RELOCATE PHASE CASE 3
SET PERCEPTOR MODE : EXECUTE
SET PATH PLANNER : BREAK LINK
SET DRIVE CONTROLLER : IDLE
CHECK PATH PLANNER STATUS
Done