COMP4034 MINI-TASK 5 MARK SCHEME

The instructions for mini-task 5 are available on GitHub: <https://github.com/RobotTeaching/COMP4034/wiki/Minitask-5>

The total available marks for mini-task 5 is 95. This will be added to the marks you got for submitting mini-tasks 1 through 5 (1 mark per mini-task) for a total of 100 available marks for the module. There are 10 bonus marks available for participation and performance in the time-trial, however the module remains capped at 100%.

You MUST attend your demonstration in person. Failure to do so will result in a failure for the demonstration component. If you cannot make your demonstration for any reason (e.g., sickness) then you must let the module convenor know as soon as possible, so a new time can be arranged.

The due date for the code component is Friday 8th December 2023. You will need to submit your code to git, tagged “minitask5”. You will also need to submit a video of your robot running in simulation (for moderation purposes) to git.

Demonstrations will be scheduled on the 13th, 14th, and 15th of December. If you wish to participate in the time-trial, you will HAVE to run your robot in the real world. We will record your time to find all the objects in the scene. Marks will be allocated based on performance, i.e., 10 bonus marks are awarded to the fastest 10% of robots, 9 bonus marks are awarded to the next fastest 10% of robots, down to the slowest 10% receiving 1 bonus mark.

The report component is due on 4th January 2024. You will need to upload your report to Moodle via the submission link in the “Assessment” section.

|  |  |
| --- | --- |
| CODE (/15) |  |
|  | INDIVIDUAL MARK: code is concise, follows good coding conventions, and is well-commented (/10) |
|  | INDIVIDUAL MARK: students' code integrates well with one another (/5) |
| DEMONSTRATION (/50) |  |
|  | The robot's code does not crash, and there are no unexpected behaviours (/5)   * There are error handling functions and unexpected behaviours are dealt with * Turtlebot may take some time to get out of narrow paths |
|  | The robot uses a suitable control strategy to manage its behaviours (/5)   * There is a pipeline that checks for map updates, it then updates new information about frontiers and centroids, publishing new markers for the turtlebot to follow as goals. |
|  | The robot finds all the objects (/5)   * Can find all the objects with the frontier exploration and output it to the terminal and rviz as a marker |
|  | The robot builds a suitable map (/5)   * The robot builds a suitable map efficiently with markers * Rviz holds a config file that provides global and local paths, alongside an energy map option to see where the robot has passed through recently |
|  | The robot localises within its map (/5)   * The robot can know its position on the map with a terminal command * It understands its position relative to its surrounding due to the frontier exploration allowing it to know where in the occupancygrid it has already explored. |
|  | The robot fails safely (you can ask to see error-handling code if the robot does not fail in the demo) (/5)   * If no map is received after 1 minute, the programs stops/fails * Auto navigation of the map will fail if the map is already explored. * If the robot cannot rotate to face its goal, it will tell the user that it is stuck, it will attempt to get out of a stuck situation up to 5 times. |
|  | The robot remembers the location of objects, and where it has been before to maximise exploration (/5)   * Adding markers to the objects explored on rviz * Terminal output of object detected location |
|  | Points available for "style and elegance": how impressed with this solution are you? How advanced are the techniques used? (/5)   * Frontier Used * Breadth first search used * HSV Detection used |
|  | INDIVIDUAL MARK: how well can the student explain their parts of the code? Do they understand how their code integrates with their teammate's? Do you feel like equal effort was put into this coursework by all teammates? (/10) |
| REPORT (/30) |  |
|  | The student has provided comprehensive, API-style documentation for their code components (/10) |
|  | The student has clearly communicated the work they did, including integration, testing, tuning (/10) |
|  | The report is well-formatted, follows a sensible order, with clear diagrams and tables where necessary (/10) |