Lincoln School of Computer Science

Assessment Component Briefing Document	
Title: CMP3103/CMP9050M Autonomous Mobile Robotics, Autonomous Mobile Robotics (M), Assessment Item One – Robotics Practical Assignment	Indicative Weighting CMP3103M: 30% Indicative Weighting CMP9050M: 20%

Learning Outcomes:

On successful completion of this component a student will have demonstrated competence in the following areas:

- [LO3] implement and empirically evaluate intelligent control strategies, by programming autonomous mobile robots to perform complex tasks in dynamic environments

Requirements

This assessment item is split into two main tasks: "Group Robot Tasks" and "Visual Object Search", both detailed below

Note: The "one-stop shop" for resources relating to this assessment component and the workshops in CMP3103M is https://github.com/LCAS/teaching/wiki/CMP3103M.

1. "Group Robot Tasks" (Criterion 1)

Your first task (relating to Criterion 1 "Group Robot Tasks" in the CRG, 30% of the assessment item one mark) consists of continuous engagement with a total of four workshop tasks you work on as a group of 3-4 students, demonstrated successfully on a real Turtlebot robot and in simulation.
The tasks will be made available to you at the beginning of each workshop session at the latest, and are to be demonstrated to a member of the delivery team by the week after the latest (this gives each team 2 workshop sessions to program, test and demonstrate each task). Demonstrating later than the defined deadline results in a reduction of the merit mark by 50%. for this individual task. This is *group work*, so it is fine to work together on the task, but only students who are present at the demonstration of the group work, and can who answer questions about it, will be awarded the marks. Absent group members will have to demonstrate the work individually, under the same rules as outlined above, ie, will only obtain half the marks if demonstrated later than the week after the workshop task has been officially released. The individual tasks are available from https://github.com/LCAS/teaching/wiki/CMP3103M.

2. "Visual Object Search" (Criterion 2 & 3)

Your second task (relating to Criterion 2 and 3 in the CRG, total of 70% of the mark for assessment item one) is to develop an object search behaviour, programmed in Python using ROS, that enables a robot to search for coloured objects visible in the robot's camera. This assessment is purely done in simulation, and *not* on the real robot. As part of this task, you must submit an *implementation*

(criterion 2) and a *presentation* (criterion 3).

Implementation (Criterion 2)

Your task is to implement a behaviour that enables the robot in simulation to find a total of 4 objects distributed in a simulated environment. You need to utilise the robot's sensory input and its actuators to guide the robot to each item. Success in locating an item is defined as: (a) being less than 1m from the item, and (b) indication from the robot that it has found an object.

For the development and demonstration of your software component, you will be provided with a simulation environment (called "Gazebo"). The required software is installed on all machines in the Labs. The simulated environment includes four brightly coloured objects hidden in the environment at increasing difficulty. Your robot starts from a predefined position. You will be provided with a "training arena" in simulation (a simulation of an indoor environment in which 4 objects will be "hidden"). This "training arena" will resemble the "test arena" in terms of structure and complexity (same floor plan of the environment), but the positions of the objects will slightly vary to assess the generality of your approach.

You may choose any sensors available on the robot to drive your search behaviour. However, your system design should include the following elements:

1. Perception of the robot's environment using the Kinect sensor, either in RGB or Depth space, or using a combination of both RGB and Depth data in order find the object;
2. An implementation of an appropriate control law implementing a search behaviour on the robot. You may choose to realise this as a simple reactive behaviour or a more complex one, eg, utilising a previously acquired map of the environment;
3. Motor control of the (simulated) Turtlebot robot using the implemented control law.

The minimum required functionality consists of a simple reactive behaviour, allowing in principle to find objects. For an average mark the behaviour should be able to successfully find some objects at unknown locations. Further extensions are possible to improve your mark in this assessment, also to enable you to find all objects. Possible extensions to the system may include (but are not limited to):

- An enhanced perception system – in-built colour appearance learning, use of additional visual cues (e.g. edges), combination of RGB and Depth features, etc.;
- Exploiting maps and other structural features in the environment or more clever search strategies.
- Utilising other existing ROS components that are available (like localisation, mapping, etc)

The software component must be implemented in Python and be supported by use of ROS to communicate with the robot. The code should be well-commented and clearly structured into functional blocks. The program must run on computers in the computer labs assigned for the workshops. To obtain credit for this assignment you will need to demonstrate the various components of your software to the module instructors and be ready to answer questions related to the development of the solution – please follow carefully the instructions given in the lectures on the requirements for the demonstration and see below for presentation requirements.

Your implementation needs to be submitted via blackboard, with the source code containing good documentation (functionality and code quality account for 40% of assessment item one).

Presentation (Criterion 3)

You are also required to submit and present a Powerpoint presentation with a maximum of *3 slides*, *including the title slide*. One slide should cover your system design, including details of the distinctive features of your perception and control algorithms. A diagram showing the system architecture is recommended for this part. The other slide should cover your results, including your observations and reflections on the system performance.

You will be given up to 5 minutes to present your slides to the module coordinators at a special workshop session (scheduled in weeks B10 and B11) and show your implementation working the labs. You must furthermore be prepared to answer any further questions about your work. Please follow carefully the instructions given in the lectures and workshops on the requirements for the presentation. For assessment, you will be assigned to a "marking group" to briefly present your working implementation and explain it to members of the delivery team to have it marked. You will be marked on both the functionality and the quality of your presentation as detailed in the CRG.

Useful Information

This assessment is an individually assessed component. Your work must be presented per the Lincoln School of Computer Science guidelines for the presentation of assessed written work. Please make sure you have a clear understanding of the grading principles for this component as detailed in the accompanying Criterion Reference Grid. If you are unsure about any aspect of this assessment component, please seek the advice of a member of the delivery team.

Submission Instructions

The deadline for submission of this work is included in the School Submission dates on Blackboard. Note that the deadline indicated is for the submission of both your implementation and your presentation slides, for the "Visual Object Search" Task. There is no Blackboard submission for the group tasks (criterion 1).

You must make an electronic submission of your presentation in Powerpoint format together with a zip file containing *all* developed code files by using the respective submission links on Blackboard for this assessment component (there will be two, one for the presentation and one for the zip file). When creating your source code zip file, please make sure you include all source files in your ROS catkin workspace (i.e. everything in the `src/` directory). Each student is required to demonstrate his or her solution to the module instructors, as per the schedule indicated on Blackboard. You must attend the lectures for further details, guidance and clarifications regarding these instructions, and make sure you approach the delivery team with any questions you might have.

Note: This is an *individual assessment* and you must not plagiarise other people's code. While it is acceptable to collaborate with your colleagues, your solution (functionality and source code) should be solely your own. Submitted source code is automatically checked for similarity. *Your implementation of the object search task is not group work!*

DO NOT include this briefing document with your submission.