UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Item Briefing Document

Title: CMP9767M – Robot Programming **Assessment Item 1 – COURSEWORK**	**Indicative Weighting:** **60%**

Learning Outcomes:

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

- LO1: critically appraise the theoretical capabilities of existing state-of-the-art robot system algorithms and components
- LO2: understand and critically appraise the requirements and limitations of robot algorithms and components
- LO3: implement and empirically evaluate algorithms and components, by programming autonomous robots to perform complex tasks in dynamic environments

Requirements

In this assessment you are expected to build an artefact using (mostly) existing ROS components. You will be using the Thorvald simulation developed at the University of Lincoln on collaboration with the robotics company SAGA Ltd to build a system that can accomplish a defined task (see below). You are expected to use state-of-the-art software engineering methodology to devise your artefact (e.g. Git-based source code management). You are assessed based on the structure and the comments of your Source Code and your performance in a 10-15 minutes viva where you show your artefact working. Your implementation will be based on Python and the ROS infrastructure.

The Task:

In this assignment you are going to implement a weeding robot. Your robot is working on a simulated farm and must drive up and down the planted rows to find the weeds and eradicate them selectively (a simple simulated spraying device is available to you in simulation). In this module you are encouraged to use any existing off-the-shelf ROS components and fragments and integrate them into your system, but you must correctly cite all used resources. The robot must navigate autonomously avoiding any obstacles. You will be provided with "training arenas" in the form of Gazebo simulations (see below for their characteristics). The actual environment your implementation will be tested in will be very similar to that "training arena", including having the same types of plants and general layout, but, the specific position of the plants on the farm will be slightly different to test for generality of your solution.

An excellent implementation solves this task as accurately and quickly as possible and is very well documented in source code. It may also feature meaningful unit and system tests to ensure high code quality. Please refer to the CRG for further details.

In order to allow you to play to your own strengths and interests, you are asked to specialise in one of three focus areas (perception, navigation and coordination). You will be provided with two different simulation environments (static and dynamic), each with two different areas with specific challenges. Please find details below.

Simulation Environments

You will be using the Gazebo simulator, providing you with two simulated Thorvald robots, and a farm world to operate in. The robots feature an RGBD camera, laser scanners, and a precision spraying device for weed treatment. There are two planting areas on the simulated farm for you to have your robots operating in:

1. "*simple*": This area features textures of only two plant types which are relatively easy to discriminate visually by the camera (e.g. by colour).
2. "*realistic*": This area of the farm features visual textures obtained from real field data sets, presenting a greater, but more realistic perception challenge.

Further to these two areas on each farm, two different configurations of the simulation are provided:

1. "*static*": In this environment there are no unexpected obstacles, and the environment is easy to navigate.
2. "*dynamic*": The dynamic environment features additional obstacles the robot will encounter on its way, which it must avoid. At least one of the obstacles is a moving human, walking across the farm.

Specialisations

As stated before, you shall specialise in one of the following focus areas:

1. **Perception**: If you focus on perception your implementation should be able to tell weeds from crops with a high degree of accuracy, and the classification and detection of weeds needs to be evaluated systematically and with respect to the state of the art. You will use the simulation environment "*static*" to show-case your single robot implementation but will have to operate in both planting areas ("simple" and "realistic") of the farm environment to show the generality of your perception implementation. Your robot must still move around autonomously.
2. **Navigation**: A focus on navigation entails operating a single robot in the "*dynamic*" environment, where your robot encounters obstacles not originally known and potentially moving around (a simulated human is walking on the farm). You may focus on the "*simple*" planting area, featuring a simpler perception solution.
3. **Coordination**: While in the other two focus areas you only operate one of the two available robots at a time (you may choose which one), the coordination focus area requires you to run both robots concurrently, to leverage the benefits of robotic fleets. You may choose to operate on the "static" environment and focus on the "simple" planting area but operating in the more difficult environments will be appreciated.

You must clearly indicate your chosen focus area in the submission of your work (in a README.md file you are required to submit with your source code). The following table summarises the requirements:

Focus Area	Simulation Environment	Planting Area	No of robots
Perception	static	realistic	1
Navigation	dynamic	simple	1
Coordination	static	simple	2

Irrespective of chosen focus area, flexibility and generality, demonstrated in the viva utilising whatever environment will be rewarded.

Should you require further explanation of the coursework assignment, you are expected to ask members of the delivery team at scheduled lectures and workshop sessions.

Useful Information

This assessment is individually assessed. Your work must be presented according to the Lincoln School of Computer Science guidelines for the presentation of assessed 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. You must submit your work as a single ZIP file via Blackboard in the respective submission area for this assignment. You must include ALL source code (no compiled artefacts), and your submission MUST COMPRISE a `README.md` file, indicating your chosen focus area and a (max 100 word) summary of your solution.

DO NOT include this briefing document with your submission.