UNIVERSITY OF LINCOLN

Lincoln School of Computer Science

Assessment Component Briefing Document

Title: CMP3641M Computer Vision and Robotics (M), Assessment Item Two – Robotics Practical Assignment	**Indicative Weighting: 30%**

Learning Outcomes:

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

- [LO2] apply computer vision techniques to solve practical problems;
- [LO4] apply advanced Artificial Intelligence techniques to mobile robotics.

Requirements

Your task is to design and develop a software component that will enable the TurtleBot mobile robot to engage in a "predator and prey" game. The software component must implement an appropriate behaviour for one of the two possible roles, either predator or prey. For an actual game a minimum of two robots, one being prey and one being the predator, are active in one enclosed room (expected to be CompLab C). The area can be considered "semi-static", e.g. walls and major features (room dividers, doors, big table) will remain static throughout the tests, while other smaller objects like chairs and other robots (and humans) will be moving. The robots are marked with brightly coloured markers mounted on the top of them in a similar way as in the provided simulation environment. There will be other obstacles in the room that the robots can exploit to hide behind. These obstacles are tall enough to completely obscure one robot from the view of another. Your design should include the following elements:

- Opponent detection component – for detecting the location of your opponent (either predator or prey);
- Obstacle avoidance – preventing the robot from bumping into objects while driving around.
- Player behaviour component – for controlling the robot according to the player type.

You should consider all possible functional and technical aspects of these three components including relevant implementation details.

The minimal player functionality consists of an opponent searching behaviour and an attack/escape behaviour respective to the player type. However, several extensions are possible to improve your mark in this assessment. Possible extensions to the system include (but are not limited to):

- An enhanced object detection system – in-built colour appearance learning, use of additional visual cues (e.g. edges), robust handling of multiple objects, global localisation, etc.;
- A behaviour coordination component (e.g. based on the Finite State Machine approach);
- Implementing both roles;
- Spatial awareness – e.g. analysing the map for good places to hide and search for hidden players, reasoning about abilities to move, robust localisation, etc.

The software component must be implemented in Python and be supported by use of ROS to communicate with the robot. The code has to be well commented and clearly structured into functional blocks. The program must run on computers in Lab C. To obtain credit for this assignment you will need to give a demonstration of your system to the module coordinator and be ready to answer questions related to the development of the solution.

Finally, the working system should be critically appraised and include thorough functionality and performance testing, assessment of the system limitations, description of possible improvements and checking the influence of the system parameters on the overall performance (e.g. robot speed, processing speed, controller values, etc.).

Report

You are required to write a technical report (max. 2500 words, over-length will incur a presentation penalty!) covering the system design and the final evaluation of the system. Your report must reference any sources, such as textbooks or web- pages, which you have used to help develop your solution.

Useful Information

This assessment is an individually assessed component. Your work must be presented according to 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.

You must make an electronic submission of your work in pdf 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 to, one for the report to be submitted via TurnItIn 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 then 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.

DO NOT include this briefing document with your submission.