COMP3631 Project: Robotic Cluedo

You are asked to implement a program that plays Robotic Cluedo! Robotic Cluedo is slightly different than the board game Cluedo. In our version, your robot needs to visit a room in an arena and identify the character and weapon in the room along with their location.

During week 11, you will demonstrate your program in Logik Space. We will create an arena for the Turtlebot, and there will be a “room” (a section with a narrow entrance) in this arena. The room will hold the image of a character and a weapon. At the beginning of your demo, you will be given a map of the environment (You will learn about what a map is, how to build it and how to use it, in Lab Session 4). Your robot will be placed at a start point, which will be the same for all groups. You will also be given (x,y) coordinates of the centre point of the room in the map. The location of images of the character and weapon in the room will not be given; your robot will have to search in the room to locate them. The room will have exactly one character and one weapon. You will be asked to report the identity (Which character? Which weapon?) of the character and the weapon. You will also be asked to report the locations, i.e. (x,y)-coordinates, of the character and the weapon.

On Minerva in the Project folder, we are providing you with a set of images of characters and weapons. During the actual demo, we will choose one character and one weapon image from among this set and place them in the room. You will notice that the images also have AR-tags on them.

Grading

In total, this project corresponds to 40% of your module grade. We will call each 1% a "point". 10 points will come from the live robot demo as a group, and 30 points will be based on your individual written report. Details of the live demo, and the written report are below.

Live Demo

Deadline: Monday, 10 December, 10:00am. (Your actual demo time during week 11 will be announced later, but your demo should be ready at the beginning of the week. You should have committed your files to your git repo by Monday, 10 December, 10:00am. No change on your git repo is allowed after this time!)

You will write a Python program to perform the Robotic Cluedo tasks. You will be graded based on whether your robot can access the room, the number of images your robot can detect, and the amount of information it can give about the images. You will also be graded based on whether your robot required any manual intervention.

You can collect 10 points, according to the following rules.

Accessing the room (max. 2 points): Accessing the centre point of the room will earn you 2 points. We will give you the x,y coordinates of the approximate centre point of the room in the map. Your robot must reach within 0.5 m of this point. Your robot will start from the designated start point.

Object search and detection (max. 4 points): Your robot is at the centre of a room. It will then search for character or weapon images in the room. The room will have one character and one weapon image, therefore in total there will be two images. When your robot detects an image, it will get 2 points. When

your robot thinks it saw the image of a character or a weapon, it should save a snapshot of the image under a directory called "detections", with a boundary around the detected image. You must also write to a file the estimated x,y position of the detected image in the map. You are welcome to use the AR-tags to detect their location. You do **not** need to specify the identity of the images to get these points. We will use these snapshots under "detections" to calculate your points after your robot is done. If there are snapshots without any actual character or weapon, you will get –0.4 point penalty for each such snapshot.

Object identification (max. 4 points): There will be 2 objects (one character and one weapon). Each identified object (Which character or weapon?) will earn you 2 points.

Manual intervention: Ideally, you will start your program, and your robot should go into the room, search, detect, and identify the images without any external intervention. However, for some reason, you may want to manually intervene this process; maybe your robot gets stuck, maybe your code crashes. During a manual intervention, you can pick your robot up, place it anywhere in the map, and re-start your program (you are allowed a minor change to your program during a manual intervention, only to specify which task the robot should resume from). Each such manual intervention will cost you -1 penalty points.

The minimum you can get from the live demo is 0 (zero) points; in other words it cannot go negative due to penalties.

Written Report

Deadline: Monday, 10 December, 10:00am.

Writeup: Write up your solution as if it was a report to a client. This should be no more than 10 sides (ideally shorter). In particular;

- Include details of the design options you considered and justification of why you chose the particular options you did.
- Describe how you have tested your solution.
- Include **images**, **a link to a video** and **data** to demonstrate that your solution works. Outline and discuss the limitations of your proposed approach. Suggest scenarios where it might not work.
- State any OpenCV/ROS codes you have used that are not part of the standard distribution.
- Discuss your contribution to the project as a member of your group.

Submission: The report is to be submitted electronically in the Minerva as a PDF file. All code should be submitted in the git repo.

Markscheme:

Design (17 points): Marks will be awarded for:

- Well designed solutions
- Justification of decisions and general knowledge of possible methods
- Novelty
- Likelihood of working in a wide range of environments and images (other than those provided)

Implementation and Results (5 points): Marks will be awarded for:

- Efficiency of planning and search methods
- Identification of object presence and location accuracy
- Identification of object accuracy

[so numbers, figures and diagrams should be presented!]

Writeup (5 points): Marks will be awarded for:

- Clarity of presentation of solution and results [N.B. Large chunks of code with no explanation are unlikely to gain high marks!]
- Discussion of the strengths and weaknesses of the system presented
- Presentation
- Self and team reflection

Self and team assessment (3 points): You should reflect on:

- Your attendance at meetings and lab sessions organized by the group.
- Your contribution to the design of the team's solution.
- Your contribution to the implementation of the solution.
- Your contribution to the testing of solution.
- Your contribution to writing up the report.
- Your contribution to the inter-personal / social well-being of your team.
- Contributions of the other members of your team.