COMP3631 Project:

You are asked to implement a program that controls a simulated Turtlebot to find and identify a Cluedo character in an environment. In the environment, there will be two rooms. Your robot needs to enter the “green room”, which has a green circle near its entrance, and identify the character in the room. The second room will be a “red room”, with a red circle near its entrance. Your robot should **not** go into this room.

You will work on this project as a group.

During week 11, your program will be tested in a demo using the Turtlebot Gazebo simulation. 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 be given (x,y) coordinates of the entrance points of the two rooms in the map. One room will have a red circle on the wall near its entrance, and the other a green circle. The green/red circles on the walls will be visible from these entrance points, but not necessarily from a direct angle. (You might or might not need to move your robot around the entrance points to have a better view of the circles. You are recommended to experiment with different positions of the circles and robot, build a robust program, and report your findings in your report.) Your robot will need to enter the room with the green circle on the door. You will be given the (x,y) coordinates of the centre points of both rooms. There will be a Cluedo character in the green room. Your robot will have to find this Cluedo character and report the identity of the character. In your group’s gitlab repo, we will provide you with a set of images of different Cluedo characters and their names. Your robot will need to identify which one is in the green room. We will also provide you with an example Gazebo environment, and associated map, and an example input file (Please see the associated file ExampleWorld.pdf to see how to use the example world). These will be just examples; the actual shape of the environment, shape of the rooms, exact size/position of the green/red circles, and the position/identity of the Cluedo character may change. Your program should be robust to such changes.

This project has three components:

- The Simulation Demo - to be implemented as a group
- The Written Group Report – to be written and submitted as a group
- The Written Individual Report – to be written and submitted individually

In total, this project corresponds to 40% of your module grade. 10% of this will come from the demo on week 11, 25% will be based on your group written report and the final 5% will be from an individual report. Details of the demo and the written reports are below.

Simulation Demo

Deadline: 17:00 on Friday, 30 April 2021. (Your group's actual demo time during week 11 will be announced later, but your code should be ready by the deadline above. You will need to push your code to your group's gitlab repo by that time, and we will use your code from gitlab during your demo.)

Submission: You will write a Python program to perform the task, which should be pushed to the gitlab repo of your group by the above deadline.

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

Detecting the green room (3 points): We would like your robot to find the green circle and save a snapshot of the camera image of the green circle (3 points). The green circle must be completely contained within the saved image. The image file name must be "green_circle.png". If an image with this name is saved and it does not show the green circle, you will get an -2 penalty point.

Going into the green room (3 points): Accessing the centre point of the green room will earn you 3 points. Going into the red room will cost you -2 penalty points.

Character identification (4 points): When your robot thinks it saw the image of the Cluedo character, it should save a snapshot of the camera image with the filename "cluedo_character.png" (2 points). The character must be completely contained within the saved image. If an image with this name is saved and it does not show the character, you will get -1 penalty point. Your program must then identify the character, by printing out the character name into a text file with the filename "cluedo_character.txt" (2 points). If a file with this name is created, but includes a wrong character name, you will get -1 penalty points.

Testing procedure: During the test, we will use your group's code that has been submitted to your git repository before the deadline. You should provide a "readme.txt" file in your git repository describing how we should run your program. Particularly, please note that:

- You will **not** be able to provide any special parameters to your program after you have seen the actual test environment. In other words, you will **not** be able to modify the behaviour of your program based on the specific environment we are testing in.
- You will **not** be able to run different scripts/programs at different stages of the demo. For example, please do not think that you can run one program to go to the centre of the green room, and there you can run a new different script. This is not allowed. You can run as many scripts as you want, but we will run them at the very beginning altogether before your robot starts to move. After we run these scripts at the beginning, you will not be allowed to input anything. Your robot should simply complete the whole task.
- We will do the test, during an online meeting with your team members. During the test, we will share our screen with you, git pull your project, run it, and mark it. You will not be using your Linux accounts. We, the markers, will be using our university Linux accounts, and we will git pull your code. Therefore, please make sure to commit and push everything you need to git, by the deadline, at which point git access to projects will be locked down.
- Here is what we will do to run your projects on the day:
    1. We will run "roslaunch turtlebot_gazebo turtlebot_world.launch" with a world file that we have created beforehand.

2. We will run "roslaunch simulated_localisation.launch map_file:=..." with a map file that we have created beforehand.
3. We will run "roslaunch turtlebot_rviz_launchers view_navigation.launch" to start RViz.
4. We will provide a "2D pose estimate" to localise the robot.
5. We will run "roslaunch turtlebot_teleop keyboard_teleop.launch", to drive the robot to the starting point in the world, and to better localise it.
6. We will kill (ctrl-c) the keyboard_teleop.launch.
7. We will run your script(s) according to your “readme.txt” file.
8. We will mark your demo according to the tasks your robot is able to perform as described above. Your robot will have at most 5 minutes after we run your script (step 7 above).

- Your robot will have at most 5 minutes to complete the tasks. If, after 5 minutes of running, your program has not stopped by itself, it will be stopped and the points you have collected up to that point in that run will be your mark. You are free to ask us to stop your program at any point and re-run it, as long as the total time spent by your robot is under 5 minutes. (We will stop the 5 minute countdown for the set-up in between runs.) **Only your last run will determine the mark you get, even if it is lower than any previous run**.

Debugging (-5 points): During testing, if you realise you want to make changes to your code, we will give you the opportunity, however this will cost -5 penalty points. If you choose to do this, we will give you access to your git repo again, and you will have exactly 10 minutes to make any changes and push your code to your git repo. After 10 minutes, your git repo will be locked again. At this point, we will ask you whether you want to test the new code in your git repository. If you say “no”, we will use your mark before you tried to debug and we will **not** apply the -5 penalty. If you say “yes”, the -5 penalty will apply and we will try running your modified program. Your robot will have a fresh 5 minute countdown. We will test with an environment that is different, to prevent you from making environment-specific changes to your code. **Only your last run will determine the mark you get, even if it is lower than any previous run.**

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

Written Group Report

Deadline: 17:00 on Friday, 30 April 2021.

Submission: The report is to be submitted electronically in the VLE as a PDF file. Only one member of a group should submit this group report. (Please see below for the additional individual report, which should be submitted by *every* member of the group separately.) All code should be submitted into the group gitlab repo.

Content: Write up your solution as a group, as if it was a report to a client. This should be **no more than 10 sides**. 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. Give examples of different environments/maps you have created to test your program, and the performance of your program in the environments we have provided.
- Include in your report **images**, **a link to a video** and **data** to demonstrate how your solution works or fails. 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.

Markscheme:

Design (13 points): Marks will be awarded for:

- Justification of decisions and general knowledge of possible methods
- Is the design structured well? Are different sub-tasks identified well? Are all of them identified? Are they integrated well?
- Was efficiency in mind during the design?
- Was robustness in mind during the design? Likelihood of working in a wide range of environments and images (other than those provided)?
- How novel is the design?

Implementation and Results (8 points): Marks will be awarded for:

- Efficiency/accuracy of reaching to the rooms, use of planning and search methods.
- Accuracy of identification of the room colors and cluedo character.
- Testing and analysis of performance, whether successful or unsuccessful.
- Use of concrete evidence (numbers, figures, tables, and diagrams) to evaluate performance.
- Video/image examples of performance.

Writeup (4 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
- Use references to credit the resources you used, if any.

Written Individual Report

Deadline: 17:00 on Friday, 30 April 2021.

Submission: The report is to be submitted electronically in the VLE as a PDF file. Each student should submit one individual report.

Content: Write up individually. Should be **no more than 1 side of a page**.

Markscheme: *(5 points)* Marks will be awarded on the basis of your individual contributions to the group work, which will be checked against what the rest of the members of your group say about your contributions in their individual reports. Please use the following criteria:

1) Attendance at meetings and lab sessions organized by the group.
2) Contribution to the design of the solution.
3) Contribution to the implementation of the solution.
4) Contribution to the testing of solution.
5) Contribution to writing up the report.

In your report, please:

- Give details of your individual contributions to the group work according to the above criteria.
- Assign a mark (between 0-5, 5 being the highest mark) to every other member of the group according to the above criteria. Provide a brief justification of your mark for the other members.