MTRX3760 - Major Project
Floods and Fires

This project contributes 30% towards your final mark. You must complete it in **groups of four or five** students, details below. Total Marks: 100.

Presentations will be held during your scheduled lab session during Week 12 for Thurs/Fri labs, and Week 13 for Monday labs. Reports are due Friday of Week 13 by 23:55. Late assignments will be subjected to the University's late submission policy unless accompanied by a valid special consideration form. Plagiarism will be dealt with in accordance with the University of Sydney plagiarism policy. Submissions will not be accepted more than a week after the due date.

Submissions will be assessed based on the following components. Incomplete submissions will have severely reduced marks.

- **Presentation**: You will present your project in-person to your peers. Your presentation will include a slide deck and also a pre-recorded video demonstrating your robot's capabilities. **All team members must speak during the presentation**. See details below.
- **Video and code**: Unlike previous submissions, **please place your video and code online**, e.g. on Google Drive, CloudStor, or DropBox, and **provide links to these in your report**. Your **code as a .zip file** (only C++ code you wrote, and no binaries) should be **separate** from your video. The video is strictly to provide a record of your system achieving the task described for this project.
- **Report**: Each group will submit a .pdf report using the class Canvas site. The pdf is the only component to submit using Canvas, so don't forget links in the report to the video and code. The report can be very simple, and needs only directly address the questions laid out in the assignment. The front page of your report should include the SIDs and tutorial sections of all members of your group.

  Code appendix: The appendix of your report **must** contain a **printout** (in text format) of all C++ source code you wrote for the assignment. File header comments and proper formatting will be critical to making this section readable.

THE UNIVERSITY OF SYDNEY

Group Work

You **must** complete this project in a **group of four or five** students.

- Each group must nominate a leader who will submit the project to canvas for the group; only one group member should submit the work
- All group members must be in the same lab section
- Projects by larger groups will be graded with higher expectations
- Groups need not be the same as for the Hello, Turtlebots project.

Background

This project takes inspiration from the NSW Smart Sensing Network's (NSSN) recent media release "Floods, fires and defence: pushing the boundaries of smart sensing through situational awareness". NSSN Ambassador at the University of Sydney and bid leader for the Smart Sensing for Australian Resilience Collaborative Research Centre (CRC), Associate Professor Sergio Leon-Saval:

> "Storms, droughts, fires and floods are examples of natural disasters that have big impacts on our communities. With more compounding disasters – droughts followed by fires and floods – resources are stretched and there is less time to recover and plan. With fire seasons being extended around the world, it's even more difficult to plan for resource sharing and we cannot continue to depend on domestic and international resources such as firefighters.
>
> **We need new ways of managing resources and multi-disaster capabilities.**"

The piece goes on to call for smart sensing capabilities and solicits input from potential partners.

For this project you prepare a bid for this program by designing, prototyping, and demonstrating a system that helps manage natural disasters using a fleet of autonomous robots. You will select a specific type of disaster recovery and focus on accomplishing specific tasks. Example opportunities are in automating the detection and mapping of risks such as fire or flood, and deploying suppression support e.g. via spraying fire or deploying sandbags.

You will prepare a 10-minute presentation explaining the strengths of your robotic system. You will present a working prototype system based on the Turtlebot 3 platform. Your prototype will be evaluated based on its functionality but also on sound system design and coding practices. The program directors know that poorly designed systems can be a liability rather than an asset. Your design should thus be extensible and maintainable, employing the object-oriented design and programming principles covered in this unit of study.

Project Requirements

Platform

Your demonstrator must make use of the physical (not simulated) Turtlebot 3 platform. You may modify the placement of sensors including the camera, and add additional hardware if it suits your project. You may not damage any part of the turtlebot.

Functionality

The application you target is up to you but must be relevant to the problem of disaster recovery in Australia. You may introduce your own idea, or pursue one of those listed in the introduction. For example:

- Automating the mapping of fires and/or floods: a robot that can roll around a non-trivial environment while sensing and keeping track of fires and/or floods in a map that it builds
- Transporting sand bags or spraying fire suppressant: it is up to your robot to find fire and/or flood sites and deliver sandbags or spray fire suppressant retrieved from a central distribution point

The idea is to demonstrate a working system design not a 100% field-ready robot. So:

- For a spraying robot there would be no expectation to build a spraying mechanism, but you might demonstrate that one would work by illuminating an LED indicating when the spray would be on in a completed robot.
- For detecting the presence of fires or floods you might consider using April tags or ArUco markers or other easily-detected markers to demonstrate the system without having to build a sophisticated detection system.

Consider building a scaled-down model of a space for the demonstration, keeping in mind that a field-ready robot would be larger and faster than a Turtlebot.

For full marks you must program substantial parts of the functionality yourself. Some suggestions:

- Autonomously move through a scaled-down space representing a disaster recovery scenario
- Identify and keep track of where fire or flood hazards exist, saving the results as a text file
- Build a map and present floods and fires on that map
- Automatically return to a distribution site to retrieve spray or sandbags, and deliver these to fire or flood sites
- Build your own navigation strategy by keeping track of where floods and fires have already been addressed and decide where the robot should go next
- Work with another team to collaboratively achieve any of the above more effectively

As a general guide, for full marks plan on implementing at least 2 of the functionalities on the order of complexity of those listed above. More grades, including up to 10 bonus marks. will be awarded for implementing more numerous or more challenging functionalities. You may also propose your own ideas suitable for the theme.

Implementation

You may employ ROS packages, libraries, and open-source code as appropriate for your system design. However, it is expected that substantial core functionalities will be programmed by you, in object-oriented C++. A design built entirely from off-the-shelf components will not receive high marks.

You may employ python or any other language to bind together parts of your system, e.g. to get the camera working. You will not receive marks for parts of the system coded in python, so the bulk of the functionality is to be constructed in C++.

Testing / Demonstration

It is up to you to decide how to test and demonstrate your system. You may collaborate with other teams in creating test scenarios and demonstration setups. To get full credit for any functionality it must be demonstrated in your video, which will be shown during your presentation and placed online.

Deliverables

You will be assessed on how well you have applied the techniques taught in the unit of study, including project management, system design, and coding skills. Refer to the rubric at the end of this document.

Presentation and Demonstration Video

During the in-person presentation in the lab you will present to your peers your proposed system design, including a pre-recorded video of your working prototype. The presentation should include:

- A clear indication of the system's intended functionality and motivation for that functionality,
- A system design overview showing how you broke the problem into modules, where each is implemented (on which processor, in which ROS node), what the class hierarchy of the main functionality is, and how the modules interact,
- A list of which functionality you programmed yourself,
- An overview of project management techniques you applied, and
- A pre-recorded video showing your robot achieving each functionality. This video need not be complex and is strictly to provide a record of the turtlebot correctly achieving the task described for this project.

Your presentation will last an absolute maximum of 10 minutes, and will be followed by a maximum 5-minute question period.

During your lab session, you will present to your peers. This is an **in-person event** and attendance by all group members is compulsory**.** There will be an overhead projector in the lab **but no speakers**. You are expected to present live, rather than relying on pre-recorded audio, and you are expected to participate actively in asking other groups questions, as well as appropriately addressing your peers' questions. Please practice presenting ahead of time, including using the projector in the lab.

Report

The report can be very brief and needs only address the following (see also the rubric below). Note that while you do list the functionalities you implemented, you do not need to show any results in your report as those will have been covered in your video. Instead, use the report to demonstrate the principles of object-oriented design and coding best-practices learned in this class.

System design

Key outcomes: Summarise what functionality you achieved. This can be a simple bullet-point list, the key results are demonstrated in your video.

Modular design: Communicate how you broke your design into modules. Be sure to show all the ROS nodes you implemented (programmed yourself) and indicate on which processor(s) each node runs. See the Hello, Turtlebots project for hints on using **ROS Node Design Diagram and UML Class Diagrams to convey your system design.** UML class diagrams need not include member variables or functions.

Class Interaction Sequence: As in Lab 1, List a set of brief bullet points that shows the sequence of class interactions followed by your robot as it carries out the **main functionality that you programmed yourself**. So for a wall-following robot this would have been the main sequence of events from when a LiDAR scan arrives until when the motor velocities get adjusted.

Hint: if one could not understand how your robot does its job at runtime from your class interaction sequence, there is likely an opportunity to increase the granularity and clarity of your design.

Teamwork & project management

Revision Control: Include in your report your git commit history as reported by:
git log --pretty=format:"%h%x09%an %ad %s" --date=short
Note: copy/paste the text, screenshots are unacceptable

Teamwork: List in brief bullet-point form the contributions of each team member to the project.

Project Management: List in bullet-point form the project management tools and techniques you adopted to coordinate the team and manage the project. Consider reviewing the Project Management lecture early in the project.

Code

As with the lab assignments, submit your code as an appendix in the report, as well in a **.zip file placed online and linked to in the report** (see front page of this handout). Only submit C++ code you wrote.

Hardware Return and Assessment

Your teaching team will be available to accept turtlebot returns in the week following Week 13, details will be posted via Canvas. You must return all parts in working condition by November 10th. Students who borrow hardware as part of this unit must return the hardware in working condition. Failure to do so will result in the final grade being withheld.

On its return, the state of your turtlebot will be evaluated, including checks for proper construction and loose parts including a loose battery. There are tools and spare parts available in the lab, so that you can rebuild / repair your turtlebot if required. There are 5 marks available based on the condition of the hardware you return.

Grading

<u>Presentation</u>	<u>45</u>	
Introduction / motivation	5	
Design overview	15	
Functionality demonstration via video	15	*+ 10 bonus available*
Teamwork & project management	5	
Presentation, organisation, clarity	5	
<u>Report & Code</u>	<u>50</u>	
System design	20	
Teamwork & project management	10	
Code quality	20	
<u>Hardware Return</u>		
Turtlebot construction, condition	5	
TOTAL	100	

Self-Assessment

[+5 Bonus] In your report estimate the level of achievement of your submission. Show estimates for all component grades as well as the total. If your estimated total is within 5 marks of the correct score you will be awarded 5 bonus marks. Ignore late penalties when estimating.