

Faculty of Engineering and IT
School of Mechanical and Mechatronic Engineering
41069 Robotics Studio 2

WaitForMe

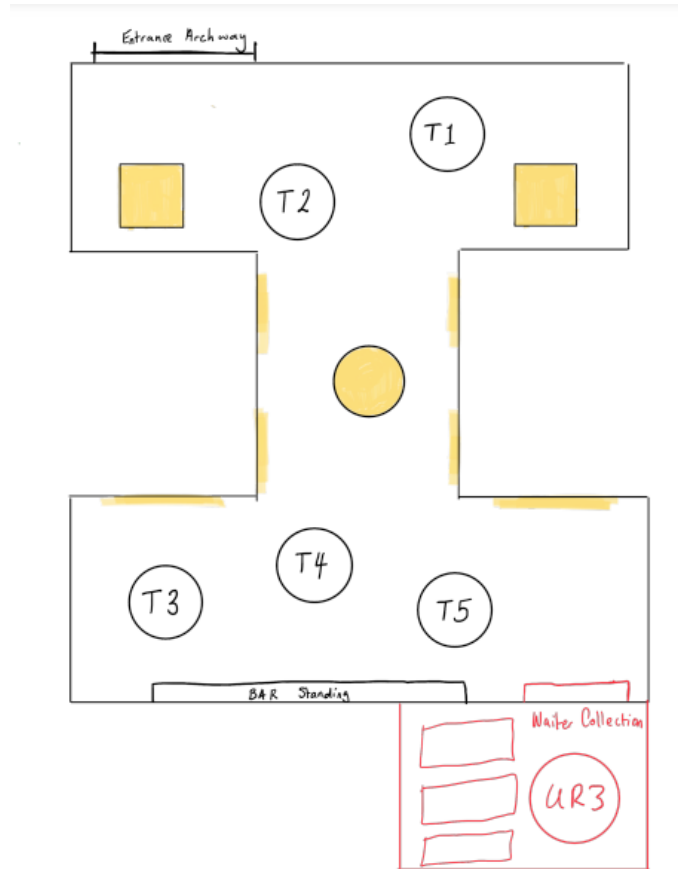
Client: Tom Ugly's Art Gallery

Project Team:

Full Name	Student ID	Expertise
Andrew Goode	13852898	Manufacturing, WSL
Issy Pitt	14040354	Rover Team Lead, Assistant Robotics Engineer, Raytheon GradX,
Hallie Robins	14253583	Systems Integration Engineer
Thomas Dodgson	13887791	C++ , MATLAB, SolidWorks, Arduino IDE

Overview

Tom Ugly's Art Gallery has tasked us with the automation of providing drinks to customer at the art gallery while they wait at tables. They have asked us to come up with a solution that utilises a TurtleBot design with multiple robots to delivery drinks in unison whilst avoiding objects and humans in the environment.



Aims

1. Create a simulated environment of the Art Gallery, tables and drinks station.
2. Simulate at least 2 TurtleBot moving between the drinks station and the tables whilst avoiding objects.
3. Simulate 2 TurtleBot's avoiding humans moving in the environment.
4. Implement functionality onto real TurtleBot's.
5. Delivery drinks without dropping or spilling drinks during transportation

Resources

Describe the resources required to complete the project. This includes: the robotics hardware & software available in the Lab, additional hardware & software you might need to source yourself, fabrication access (3D printing, laser cutting, etc.).

- Minimum of 2 TurtleBot's for use in the lab
- ROS2 and gazebo
-

Subsystems and Responsibilities

Subsystem	Description	Lead by	Supported by
SLAM, Path Planning and Object Avoidance	Create logic to ensure object avoidance based on laser scan data, create a map to ensure localisation and optimal path planning in a known environment	Andrew	Tom
TurtleBot Movement Logic	Logic includes travelling to selected tables based on information obtained from the host robot, and after delivering drinks, returning to the drinks station.	Issy	Hallie
Simultaneous TurtleBot Operation	Create logic to allow two TurtleBot's to move without colliding with each other, and they will follow the same logic as above without duplicating movements (e.g. going to the same table.)	Hallie	Issy
Software to Hardware Integration	Transfer logic successfully into two real TurtleBot's that will mimic our simulation and maintain functionality	Tom	Andrew

Evaluation

Additional information and clarification can be added here.

Subsystem 1: Path Planning, and Object Avoidance	
P	TurtleBot's can reach goals without colliding with any stationary known obstacles
C	P and TurtleBot can recognise and navigate around newly discovered stationary objects.
D	C and execute optimal path planning even if the route is interrupted by stationary obstacles that were not previously mapped using laser scan
HD	D and TurtleBot's and can ensure it will not run into any other moving TurtleBot's

Extension	HD and TurtleBot can navigate dynamic environments and avoid multiple actively moving obstacles and still reach end goal.
Subsystem 2: TurtleBot Movement Logic	
P	TurtleBot moves to a hardcoded table destination and returns to hardcoded drinks location
C	P but TurtleBot receives goal data from host for destinations, but goals may overwritten when new ones are published or node does not listen and save new goals during actions.
D	C and TurtleBot can navigate to specific locations given by host and return to drinks stand, repeating action indefinitely/always listening for new goals from host
HD	D and the logic is scalable for multiple TurtleBot's to go to all of the given goals and return to drinks table
Extension	HD and TurtleBot can navigate dynamic environments without losing functionality and still getting to the goals
Subsystem 3: Goal Management and Assignment	
P	Receives communication from publishes and subscribers. Organises and distributes information to 2 different TurtleBot's. both robots receive the goal and complete the same action.
C	P and having two moving simultaneously with the same driving logic implemented and receive an unlimited number of unique goals.
D	C and optimising which data is sent to which TurtleBot before homing based on time to Home
HD	D and goals can be executed in non-chronological order to handle urgent or priority tasks.
Extension	HD and Logic is scalable to a total of 5 turtle bots
Subsystem 4: SLAM	
P	Can localise itself on a map in a known environment
C	P and can distinguish between tables, people, and artwork from LIDAR data (use opencv feature detection)
D	C and can distinguish the turtlebot and track its movements with self-written function.
HD	D and distinguish objects without using a library (eg use ML such as SVM)
Extension	HD and create own mapping logic (.cpp)