**Turtlebot software for schools outreach.**

**Background**

A turtlebot has been created to be used by beginner programmers at outreach events. It can draw lines and pictures by utilising the robot’s various commands, including moving forward, turning, and pen controls. A simple, user-friendly GUI would be beneficial so that users can easily control the robot.

The overall intention is to bring these turtlebots and the software to outreach and open days hosted by the department. It can then be used by potential students and others to have a go at coding to control the robots. These students are assumed to be new to programming, so the language they code in needs to be relatively simple, and instructions should be provided to help.

**Aims and objectives**

The project aims to produce an application with a simple graphical user interface that can be used to type in code and control the turtlebot. This is with the intention to provide an interesting activity to do at outreach/open day events.

The application should let the user type in code and choose whether to run it on the turtlebot or via a simulation on the application, or both. Both should work with the same user-entered commands. To do this, the user's typed commands will need to be translated into a format that the turtlebot can understand before sending them.

The application should be able to provide the user with information on what commands they can use. It should also let them configure some settings, such as the amount the pen needs to go down by to make contact with the paper.

**Tasks**

* Understanding the pre-made turtlebot commands.
* Look into ways to compile and run files dynamically.
* Design a graphical user interface.
  + Have a space where users can type in code with syntax highlighting.
  + Have a turtlebot simulation visible on the screen that moves in accordance with the code typed by the user.
  + Create a settings area where the user can adjust factors such as minimum and maximum pen height.
  + Show the user the possible commands they can type, most likely with an in-built manual.
  + Create a stop button that can stop the turtlebot and the simulation.
* Investigate options for connecting to serial ports so the turtlebot can be sent messages, and they can be received from it as well.
* Have the user’s written code be able to move the turtlebot.
* Have the software be able to read in a configuration file for the turtlebot and forward it to the turtlebot, this will be for factors such as wheel diameters.
* Application testing
  + Primarily manual testing with a test table, which shall be in the report.
  + Some unit testing where appropriate in the code.

**Deliverables**

The project will deliver a GUI application that allows the user to enter code and run it via the TurtleBot and/or a simulation.

**Tools/Hardware/Software/Datasets used:**

* Python with the Spyder IDE
* Turtlebot