CSCE 274 Section 1 Fall 2016 – Project 01 – Boyd Compton, Jose Tadeo, Timothy Senn

Project Description:

In regards to the interfaces for tasks one and two, an instantiable class was used to encapsulate all the complexities of a serial connection and the iRobot Create 2. To use either interface, first, instantiate an instance of the class, and then, issue any of the available methods. Any constants associated with the specific interface were also included in the module. For example, when a command is sent over a serial connection, it needs to be encoded as a character. To encapsulate this specific complexity, the method that sends commands will encode a space delimited string before sending the command over the serial connection. This decision increased the usability of the serial interface by making the command passed to the method readable. Similar considerations for the user experience and readability were made for other areas of complexity such as the drive command, reading a button, and connecting to the robot. The interfaces also used default values for a method's parameters whenever it made sense. For example, when instantiating an instance of the iRobot Create2's interface, the buad rate will be defaulted to 115200, and the timeout will be defaulted to 1 second.

For the main program, the main while loop will check for the clean button to go from a released state to a pressed state. If this condition is met and the robot is stopped, then the main loop starts moving clockwise along a square with 250 mm sides. This is done by creating and starting a thread that is responsible for controlling the actuator. However, if the robot is stopped and the clean button is pressed, the main loop will stop the robot by stopping the spawned thread. The main loop will also remove the reference to the expended thread at this time. The actuator controls for the square are comprised of the drive forward and turn 90 degrees clockwise. The time to wait between actuator instructions is figured out by dead reckoning. In this instance, the robot knows its supposed velocity and the distance it needs to travel.

Project Evaluation:

The interfaces work as expected. They increase code clarity and readability while also encapsulating the complexities. The serial connection interface is capable of opening a connection, closing a connection, sending commands, and reading raw data. The iRobot Create 2's interface is capable of controlling the state of the robot, reading the state of the button, and sending drive commands. The different state, buttons, and special cases for radius are also included as constants in the module. When one of these methods is called, the constants can be used as arguments to increase the readability and clarity of the code. These methods also encapsulate the complexity associated with serial connections and the iRobot Create 2. Thus, these interfaces achieve their goal.

The main program will work reasonably well given enough clear space, a flat surface, and a low amount of friction. The main reason for all these environmental constraints is because the robot's navigation is based on dead reckoning. If the surface is not flat or the friction is not negligible, then the final position will differ from the desired position. This will introduce an amount of error per side of the square. Thus, the error can build up if the environment does not meet the constraints. The best case scenario is when the robot follows a square with sides less than 250 mm in length. The reason for the clear space requirement is because there is no obstacle detection or avoidance included in the robot's navigation of the square. Overall, the main program will work well in lab 3D22's environment.

Allocation of Effort:

Boyd Compton:

- Created the serial connection and iRobot Create 2 interface
- Helped write the main program for task 3
- Aided in debugging the interfaces and main program
- Helped document the interfaces
- Wrote the report

Jose Tadeo

- Helped write the main program for task 3
- Debugged the interfaces and main program
- Helped document the interfaces and main program

Timothy Senn

- Debugged the main program.
- Ensured consistent documentation and coding style for the interfaces and main program
- Wrote the README.txt