**Search and Rescue Library**

This library contains methods that will be useful for the Lego Mindstorm Search and Rescue activity. Examples include methods that can be used to follow a black line, make turns at intersections, turn around, detect obstacles, and detect a person that needs rescuing.

* **StartLineFollow() –** Begins the line following code. This needs to be called once, at the beginning of the program. None of the other methods in this program will work until this has been called.
* **StopLineFollow() –** Stops the line following code. The motors of the robot will be turned off.
* **TurnRight() –** At the next intersection the robot will take the path going to the right.
  + **TurnRightAndReturn()-** Same as above, but the function returns immediately instead of waiting until the turn is made.
* **TurnLeft() –** At the next intersection the robot will take the path going to the left.
  + **TurnLeftAndReturn()-** Same as above, but the function returns immediately instead of waiting until the turn is made.
* **GoStraight() –** At the next intersection the robot will take the path that continues going straight.
  + **GoStraightAndReturn()-** Same as above, but the function returns immediately instead of waiting until it has crossed an intersection.
* **WaitUntilTurn() –** Before any other instructions can be carried out, a turn must be made. Any code after this statement will not be executed until the robot makes a turn.
* **WaitUntilPress() –** Before any other instructions can be carried out, the robot must bump into something. Any code after this statement will not be executed until the robot touches something.
* **WaitUntilObstacle() --** Before any other instructions can be carried out, the robot must detect an obstacle. Any code after this statement will not be executed until the robot touches something.
* **TurnAround() –** Rotates the robot 180 degrees. This is useful for when there is an obstacle in the way, or after you’ve found a person and you don’t want to run them over!
* **CheckForObstacle() –** Returns whether or not there is an obstacle in front of the robot.