April 5, 2016

Michael Dunleavy, Jack Mc Cormack

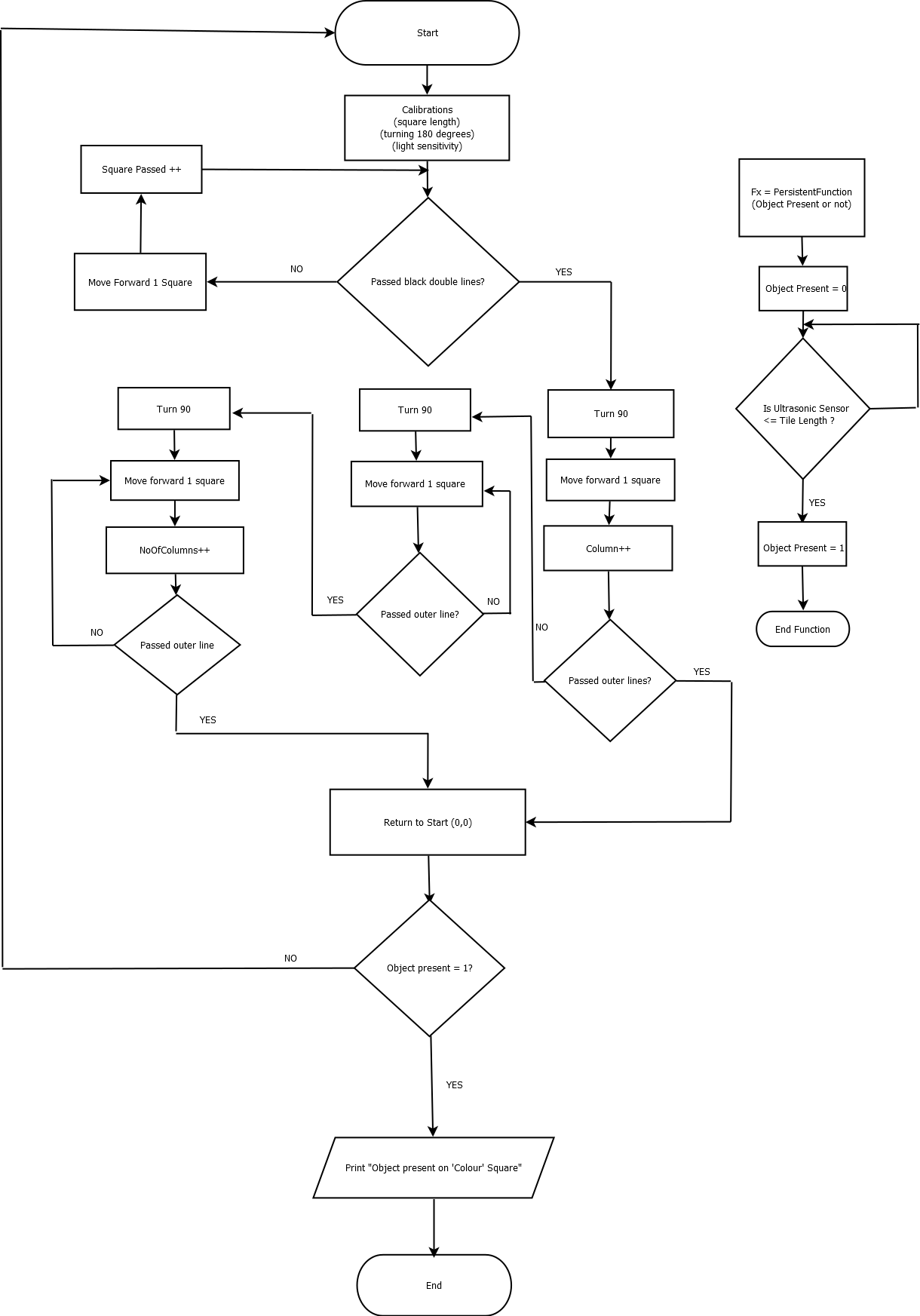
Design Document

Team Computing

Program Description

* Calibrate a light sensor threshold.
* Calibrate the length of one grid tile.
* Calibrate (using the motor encoders) how far the motors will have to turn to complete a 360° point turn.
* Place robot on starting position.
* Move forward until a double line, measure distance travelled and return to the starting position.
* Move backword until a double line, measure distance travelled and return to the starting position.
* Convert the sum of these two distances (the width of the grid) from centimetres to tile lengths.
* Calculate starting position’s x co-ordinate (in terms of tile lengths.)
* Repeat above steps for the y axis.
* Write these dimensions, the starting position, and the area to map txt file.
* Declare a 2D array with the dimensions we’ve just measured to represent the grid.
* Move to tile (0, 0).
* Declare two counter variables for the number of black and white tiles.
* Using a ‘move to tile’ function that takes x and y co-ordinates, and a ‘rotate n degrees’ function, traverse the grid in the following snaking path:
* The robot’s co-ordinate and orientation at any given time will be stored in variables and will be updated by these functions.
* Take light readings at distance intervals of one tile length and store a corresponding value in the appropriate array element. Increment the appropriate colour counter variable.
* Once the grid has been traversed, write the 2D array to the map txt file along with the colour counter variables.
* Return to the starting position.
* Traverse the same path with the ultrasonic sensor set to stop the robot when an obstacle is detected one tile away.
* Write the obstacle’s location to a second text file.
* Read from the map array whether this is a white or black tile and write this information to the second file.

Flowchart of Program Logic



|  |  |
| --- | --- |
| Task Plan |  |
| Week 9 | Collaboration to agree on the top level design of the program and how it will complete the necessary objectives. This process will involve agreeing on what functions we will need and then prototyping these functions.  Completion of the design document with each team member taking two of the four sub-tasks.  Complete weekly journal entry. |
| Week 10 | Delegation of function design after agreeing on design of arguments and return values. Most functions will fall into one or more of the following categories:   * Motion * Sensing * Data processing and encoding   Implementation of said functions. This can be commenced outside of lab time and tested during labs.  Complete weekly journal entry. |
| Week 11 | Main program implementation. This will be constructed with the functions written during week 10. This will be divided into 3 subroutines:   * Initial calibrations and measurements * Mapping the grid * Locating the obstacle   Complete weekly journal entry. |
| Week 12 | Testing and debugging. We have major concerns about rotation accuracy due to the lack of a gyro sensor. We have allocated a significant portion of time for this and other concerns. However, we are limited to the allocated lab time.  We also want to test alternate hardware configurations to see if accuracy can be improved. Specifically, a four wheeled layout.  Complete weekly journal entry. |
| Week 13 | Final testing of the robot during lab time.  Demonstration of the completed program.  Authoring of the final report as a team, and the peer review report as individuals.  Complete weekly journal entry and submit completed journal. |

Robot Build

The build will be based the standard REM build we have been using in prior labs, pictured below. (Picture courtesy of Carnegie Mellon University.)



This build will have differential steering powered by two independent motors. One turns the front left wheel. The other, the front right. A third caster wheel is mounted on the back for balance. However, we do wish to test four wheel layouts to see if it will improve turning accuracy.

We will have three sensors:

* A front mounted light sensor to read the grid and allow traversal.
* A side mounted touch sensor which will be pressed to initiate sequences of the program.
* A top mounted ultrasonic sensor to survey for obstacles.