Robot Maze Solver – Programming Things Assignment

Strategy:

My robots strategy is to check both left and right of it to see if there is a gap, if there isn’t it attempt to go forward and repeat until there is a gap or the forward hits a line. If a there is a gap the robot will go forward about the width of the door to see if it’s a corridor or a door that it has found. If it’s a door it will reverse half way back, turn 90 degrees and go into the room. Once in the room the robot will slowly spin and check for objects in the room, if one is found it will log it and leave the room and carry on down the corridor. Once the robot reaches the end (signified by a dead end for my robot) an optimised root is made from all logged moves. Only right, left and room entries are logged. The robot will then use the optimised path to determine its moves, if left, right or back its will only go forward until it hits the end of a corridor and then will perform the correct turn if any. If a room is the move it will only check the wall with the door. Once all action on the path has been complete the robot will come to a halt.

Break down:

To move down the corridor I wrote a function called moveForwardInInches(int) this method moved the zumo forward in inches based on the values passed to it. The move is based on the zunits example in the maze solver sketch. At each iteration of movement I use another function called checkSensors() which returns whether or not the zumo is currently above a line. It loops through and uses the ABOVE\_LINE macro which is also borrowed from the maze solver sketch.

To turn the Zumo I have written a function named turn(float). The function relies on the compass to get the average heading of the zumo and checks that the heading equal to the desired heading, if it is not it continues to rotate. The code borrows the turn code from the Compass example sketch along with the heading and averageHeading() code and the compass initialisation code.

I have two checks in place for left and right that checks there is a line either side of the zumo. The checks use the turn method and turn by 60 degrees and then use the moveForwardInInches method to move and check for a line. Once the check is complete the turn method is used again to turn back straight.

If a gap has been detected the checkForDoors() function is called and will move the length of the door checking for a line (wall in front of it) as it goes along. If none are found it assumes it has found a door and will reverse back half way and turn 90 degrees using the turn method and will enter the room.

Once inside rooms the spinAndCheck method is called which slowly turn the zumo and uses the scanForObject method in between each turn to try to detect an object. If once is found the scanForObject will return true and will stop the zumo and make it turn back towards the corridor and will leave the room.

I am using an integer array to store all my moves which are set to values in an enum which use left, right, leftDoor, rightDoor as its values. Only corridor turns and occupied rooms are logged. Everything else is assumed to be straight.

Once a dead end is detected the mazeReturn( ) method which start the final loop to return to the start. It uses the optimiseReturnRoot() method to convert the current path in the correct return path. Once the path has been created it enters a for loop which will loop through each command until they are completed.

Issues:

Originally I had attempted to just turn using the motors and delaying by a specific amount however this was very inaccurate and was effected by many variables such as battery charge, surface frication ect. I then started to use the compass to try and turn more accuratly however this also had its issues and its was easily scew by interferance. To minimise the affects of this i would log the current heading and only change this when i turn for a corridor. I would then turn to this heading a few times to get closer to the actuall heading i needed.

Once i attached the sheild the compass would stop working, to fix this i had the bridge 2 pins that weren’t present on the sheild but where on the arduino, the sda and scl pins.