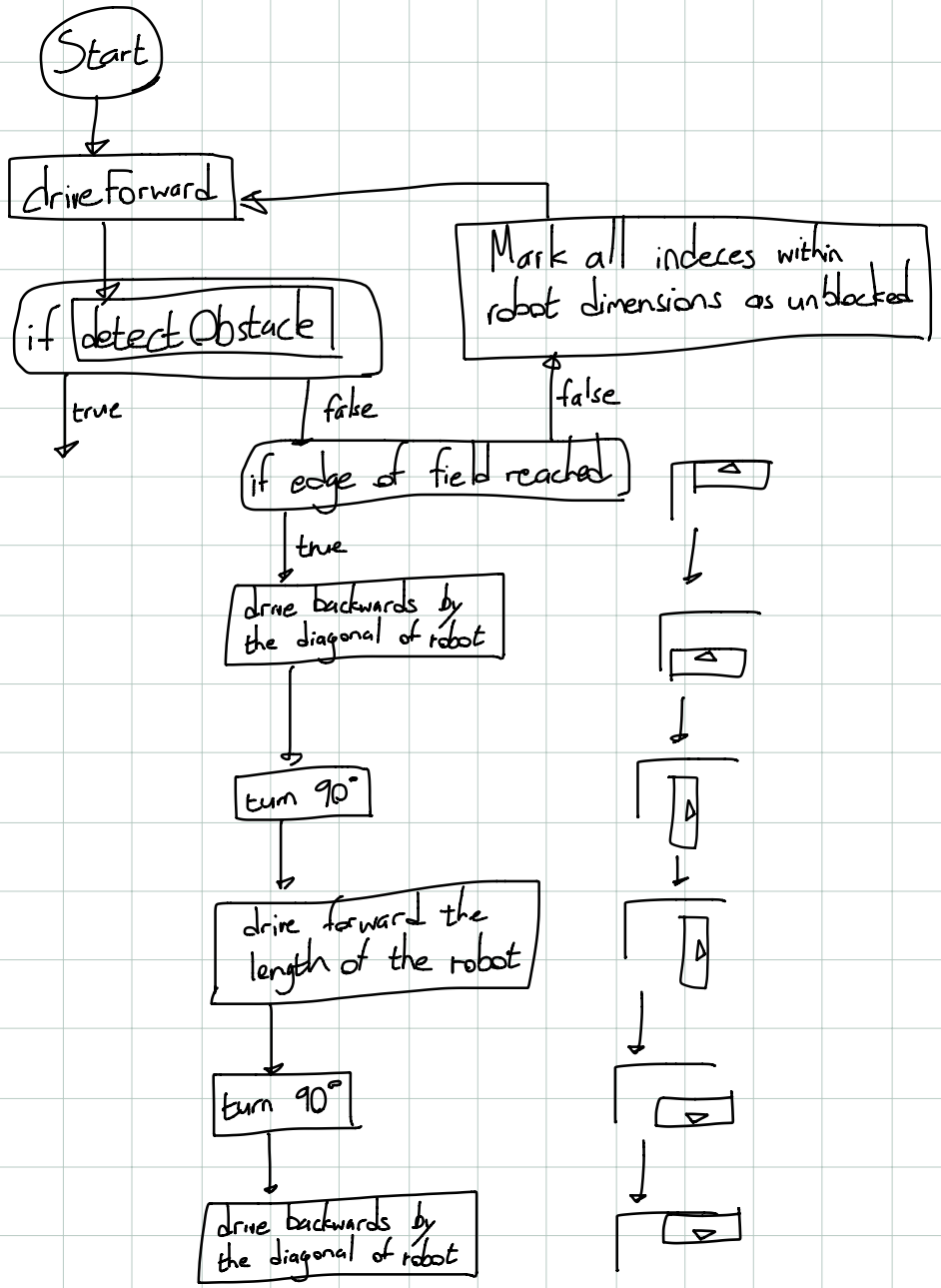
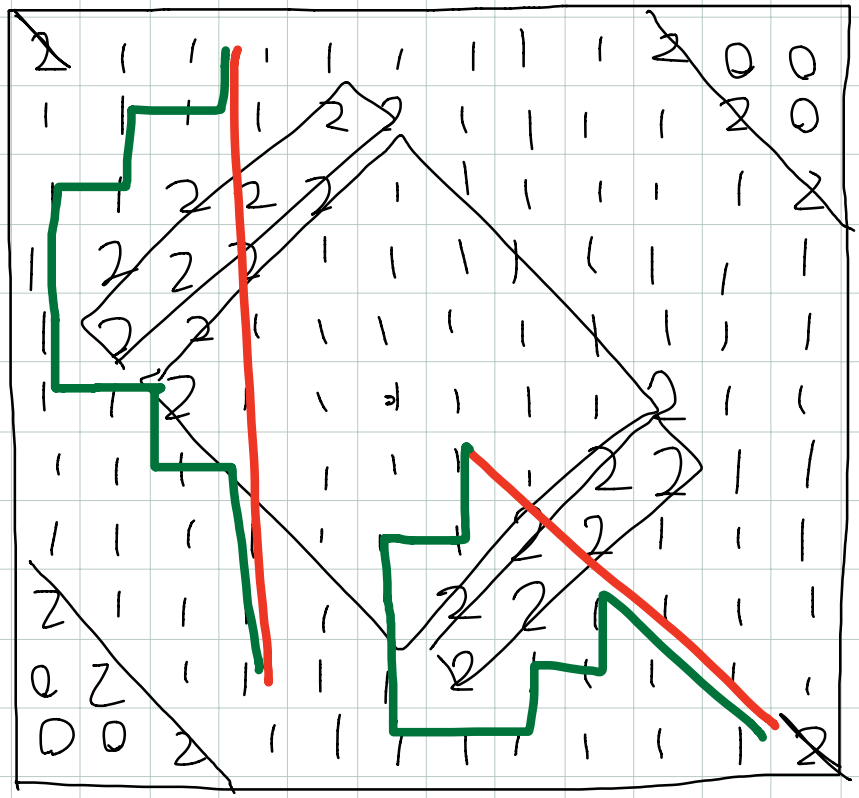

Start
driveForward
if detectObstacle
true
false
Mark all indeces within robot dimensions as unblocked
false
if edge of field reached
true
drive backwards by the diagonal of robot
turn 90°
drive forward the length of the robot
turn 90°
drive backwards by the diagonal of robot

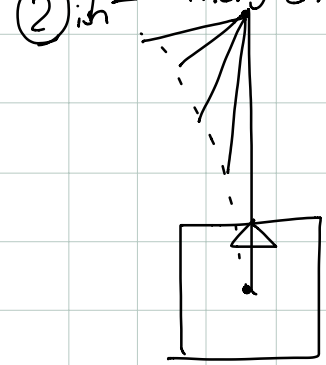
Example of field array



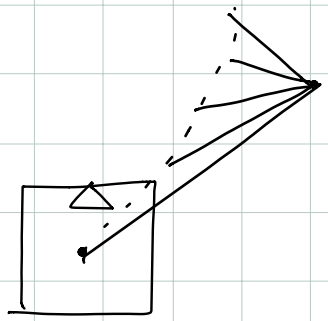
Things we must know how to do:

1. How to identify an index as 0, 1, or 2 (obstacle detection)
2. Differentiate between an obstacle and drift
3. What to do when an obstacle is found before and after field array is constructed
4. Maintain accurate location tracking on ramps - suspension + gyro
5. Pathing around know obstacles

② ish ← finding drift



currentθ should = goalθ



Expected vs. Actual

expectedPosition vs. CurrentPosition

θ $|goalAngle - currentAngle| < 0.005$

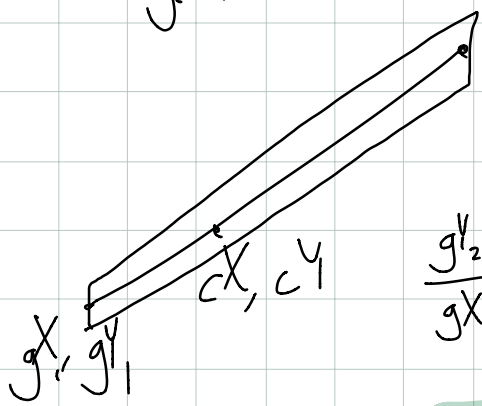
x

y

goalX

goaY

goalθ



$$\frac{g^Y_2 - g^Y_1}{g^X_2 - g^X_1}(c^X - g^X_1) = (c^Y - g^Y_1)$$

$$\left|\frac{g^Y_2 - g^Y_1}{g^X_2 - g^X_1}(c^X - g^X_1) - (c^Y - g^Y_1)\right| < 0.05_{cm}$$

① Sudden speed drop = obstacle hit

③ ~~Before array construction:~~
~~Back up, turn right, keep going~~

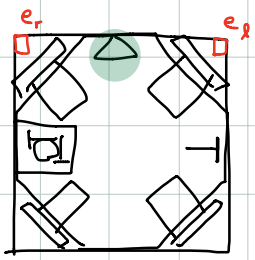
After array construction (most likely a robot has been encountered)
Back up,
use EOPD to find most open space,
drive right/left a little
drive in direction of EOPD for 18 in (unless obstacle encountered)
resume goToPosition commands

⑤ ~~if point along found to be a 2~~
Maybe the field array complicates things too much

Cons:
Need to create field array
Doesn't help with robot detection

Pros:
It's cool
Is the building block to autonomous in teleOp

The new Entach robot:



- Tracks x, y, & θ
- Drift correction with expected vs. actual ($m(x-x_1) = y-y_1$)
- Speed trackers for all