Pseudo Code

2 parts (2 different funcitons?):

1. Choosing destination
2. Choosing path to the chosen destination

Assumptions we make:

1. Enemy robot is named Tony
2. Tony is coming at us
3. We don’t trust that Tony is going in any specific direction (his direction can change)

Choosing destination:

* We want to maximize distance between Tony and the destination we are aiming for
* Unless we are within some range of Tony, then we want to bail (might not be necessary with part 2)

Choosing path:

* Consider every path to the destination
  + Break the map down into a matrix of costs
  + Don’t need to consider paths that go through or near Tony
* Weigh the cost of each grid in matrix based on its proximity to Tony

**Option 2:**(for speed)

1)Only at the paths behind near you in a circle.

**Option 3:**

* Break the map down into a matrix of costs
* Take gradient of cost vector
* Use Gradient of matrix to calculate minimum path