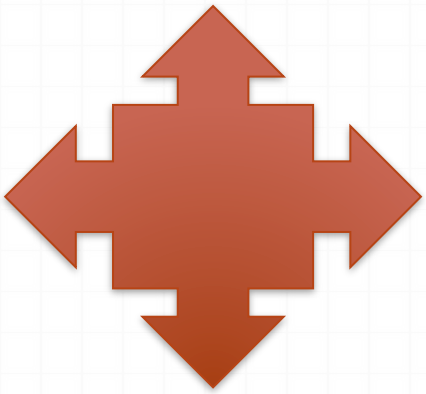
Multicopter Gils Camp

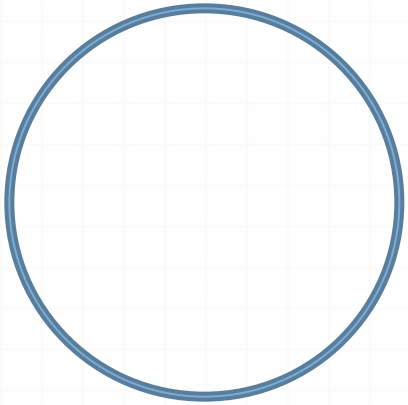
Iaroslava Grinchenko
Katarina Borovina
Aygun Ilkhaszadeh

Problem Description

Used symbols

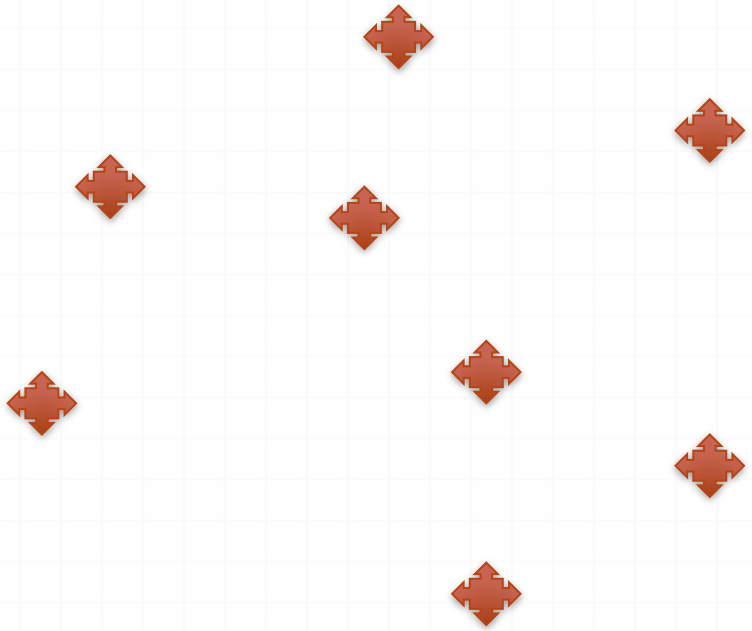


- Multicopter



- Enclosing circle

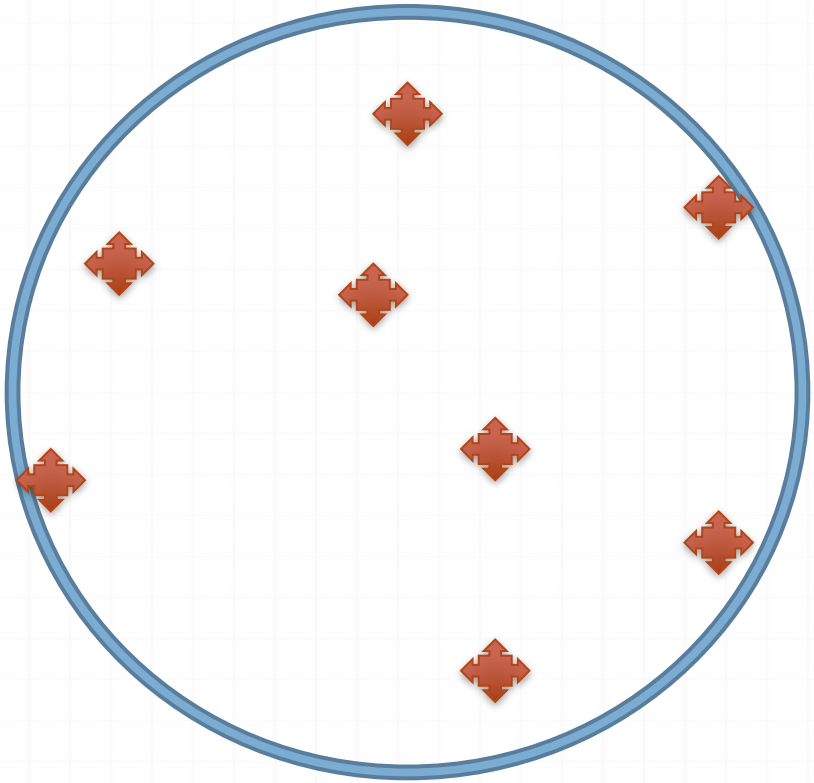
Initial setup



Algorithm *

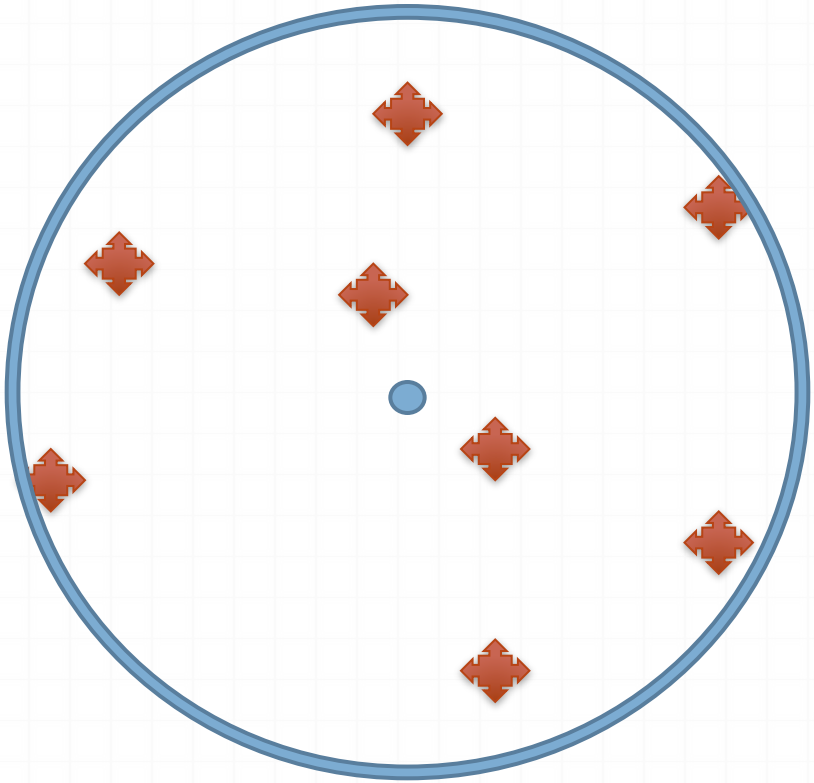
* Adjusted algorithm from the paper A.Gautam, S.Mohan “A practical framework for uniform circle formation by multiple mobile robots”

1) Smallest enclosing circle[1]



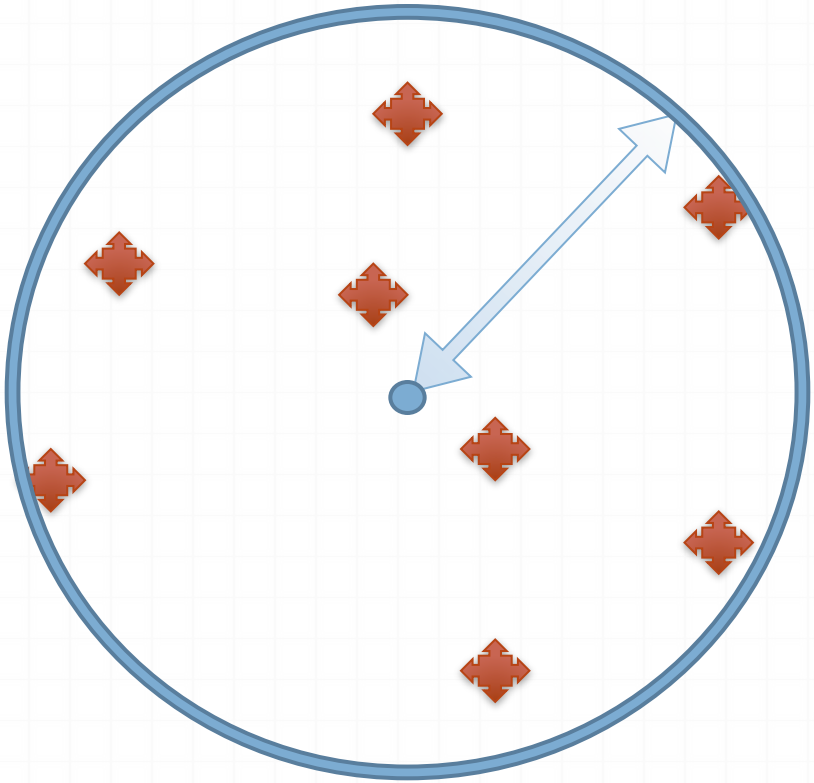
[1] Algorithm described in the paper Skyum, S.: «A simple algorithm for computing the smallest enclosing circle»

2) Identify its center point[2]



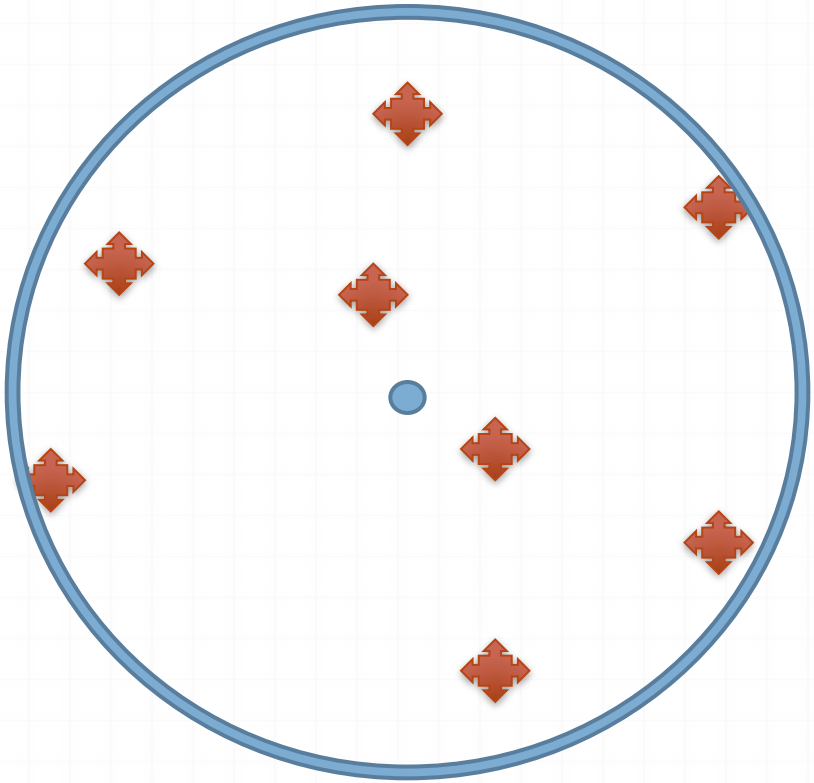
[2] let us denote this point as Center

3) And radius[3]

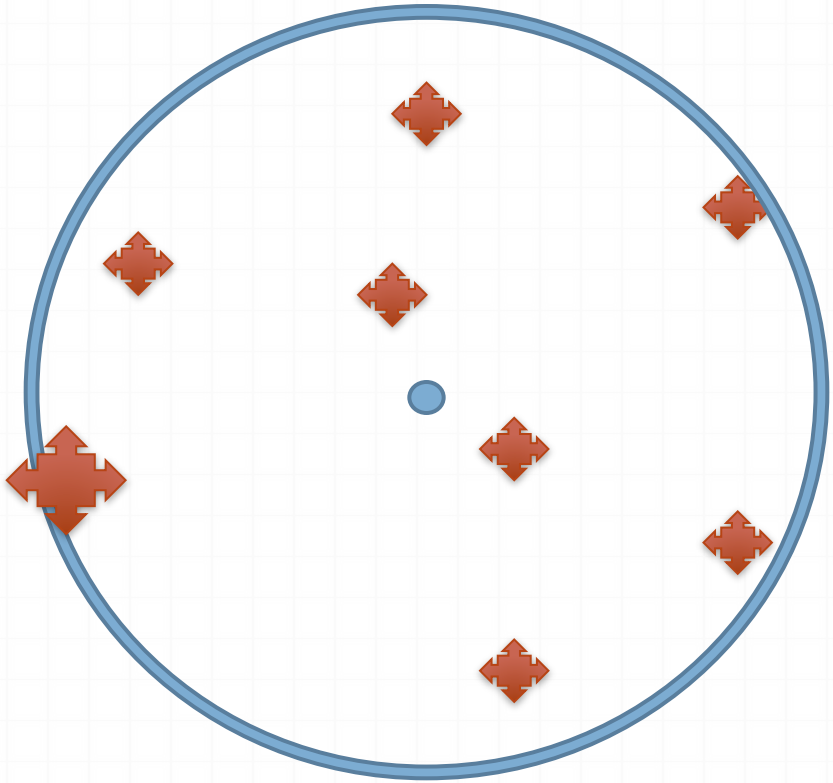


[3] R – radius of enclosing circle

4) Make center point a «Leader» or a «Mass center»

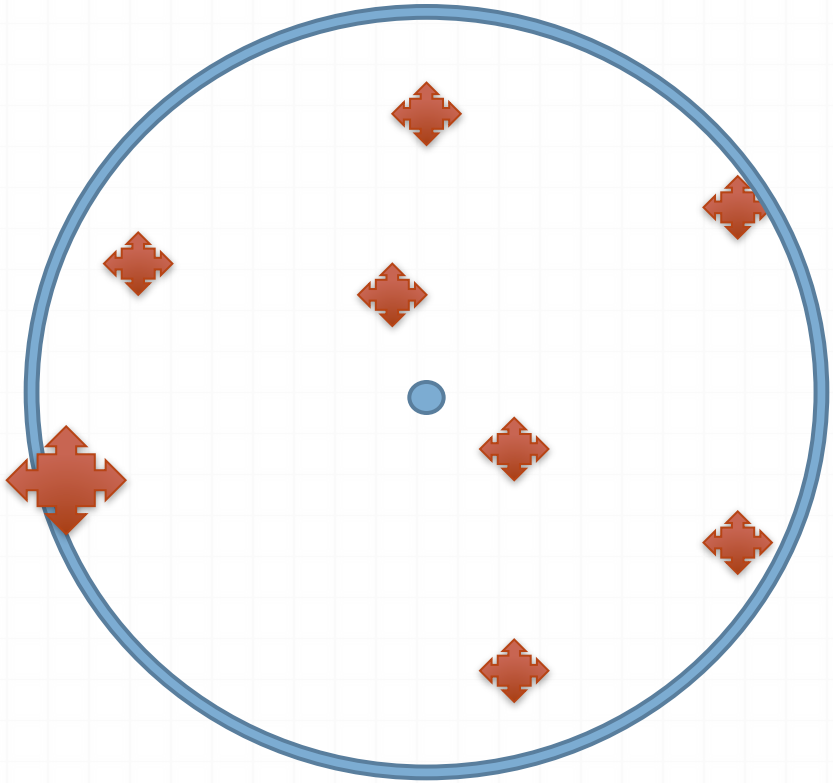


5) Detect the robot on the enclosing circle[4]



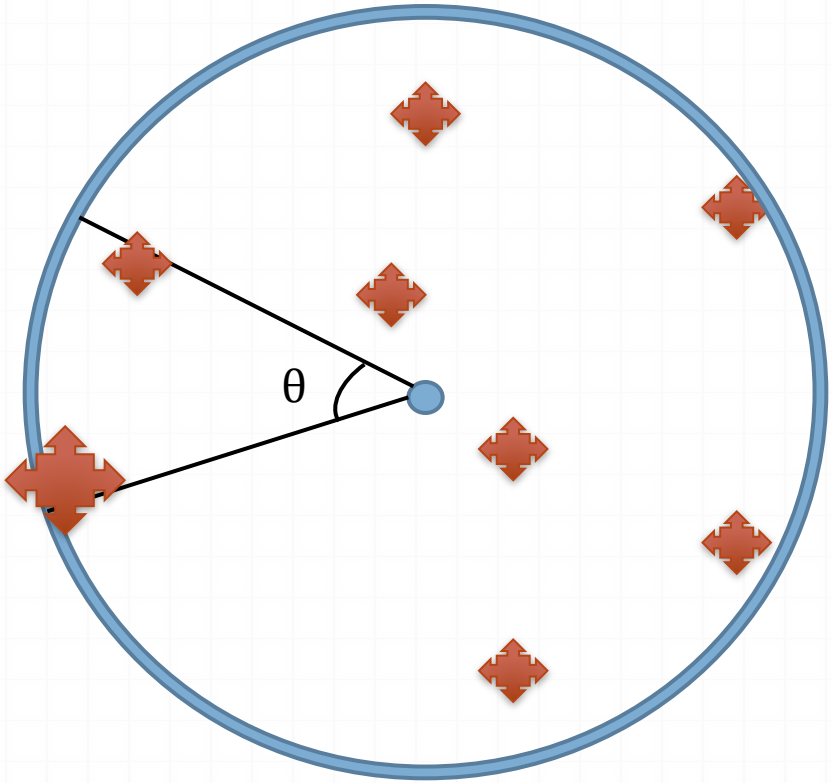
[4] let us name him Robot1

6) Count a number of robots[5]



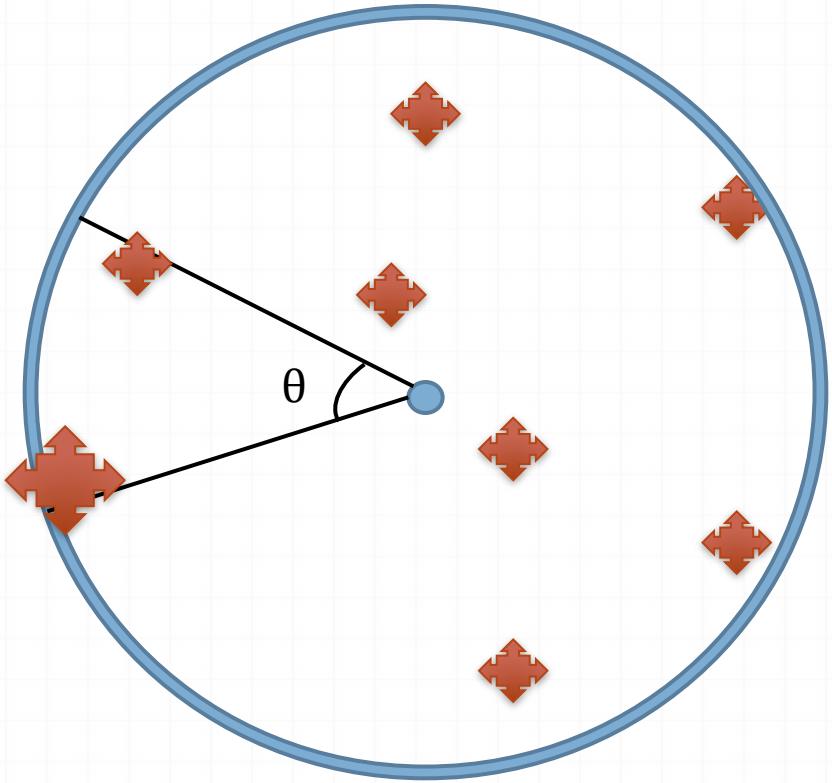
[5] N – number of robots. In our case N=8

7) Count a needed angle between robots[6]



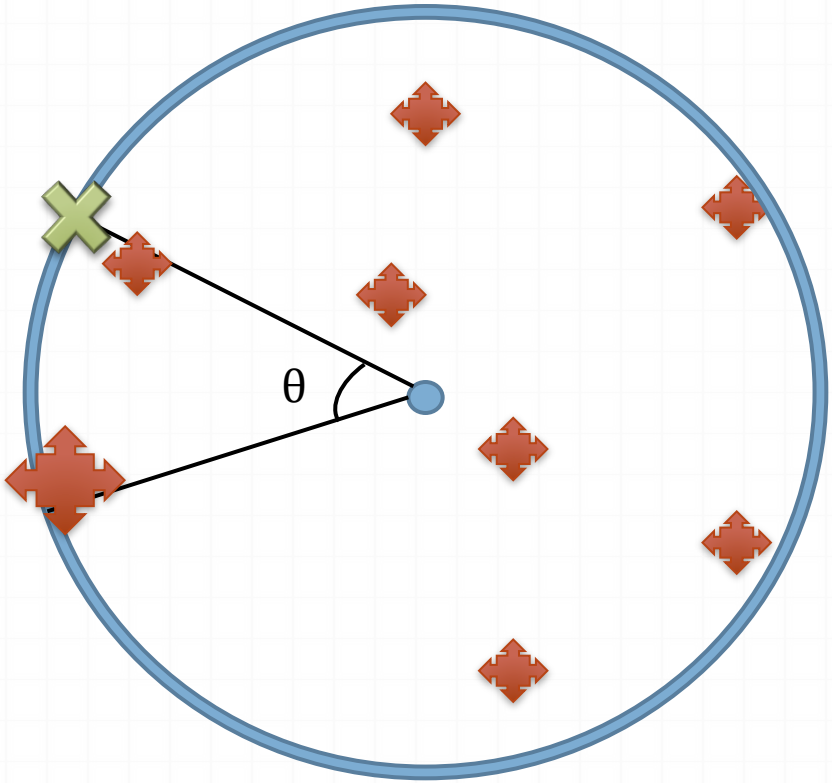
[6] θ – estimated angle between robots. θ = (2*π)/N

8) Compute relative position[7]



[7] x = Robot1.x – Center.x
y = Robot1.y – Center.y

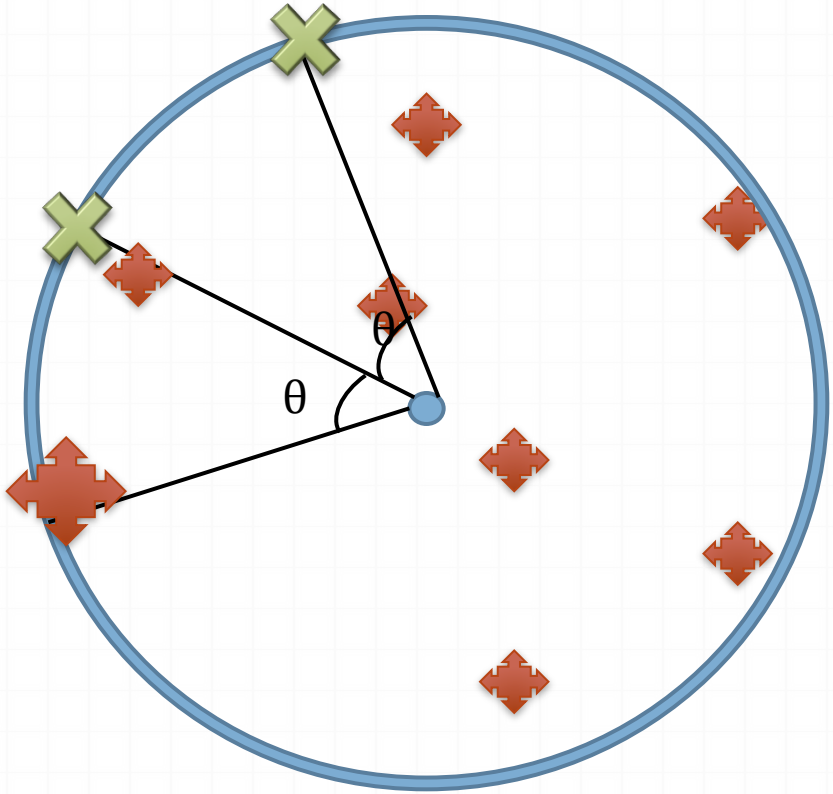
9) Compute position of all cross points[8]

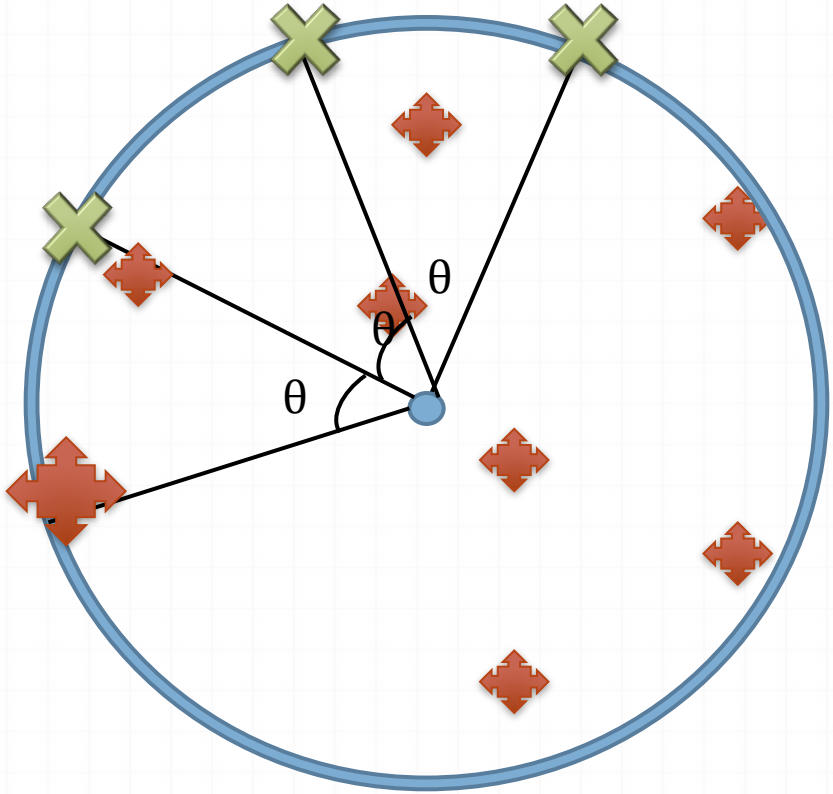


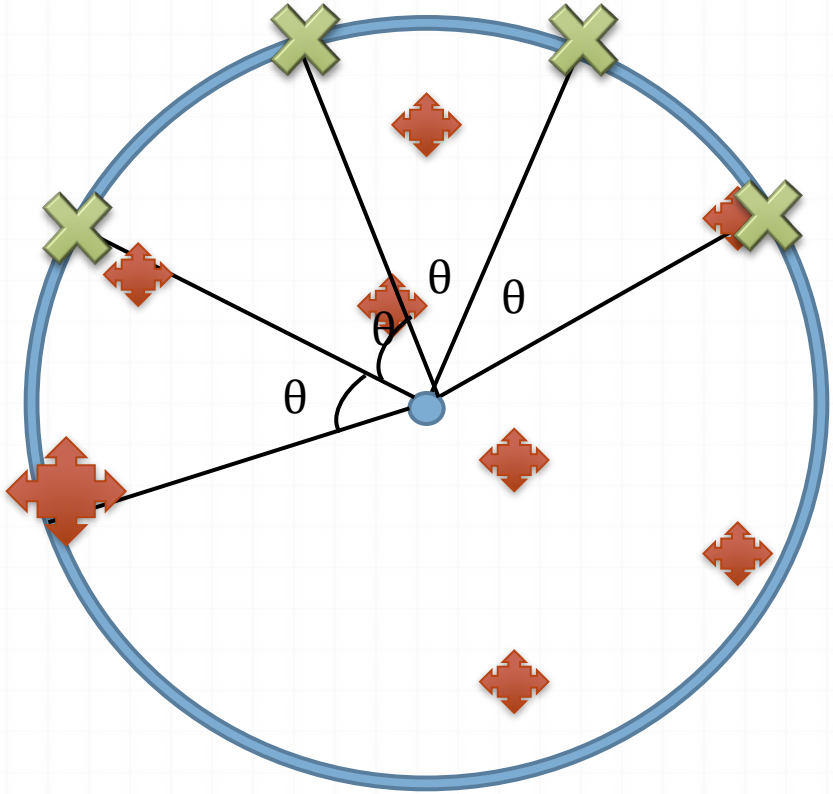
[8] Iteratively compute for all i = 1 to N-1:

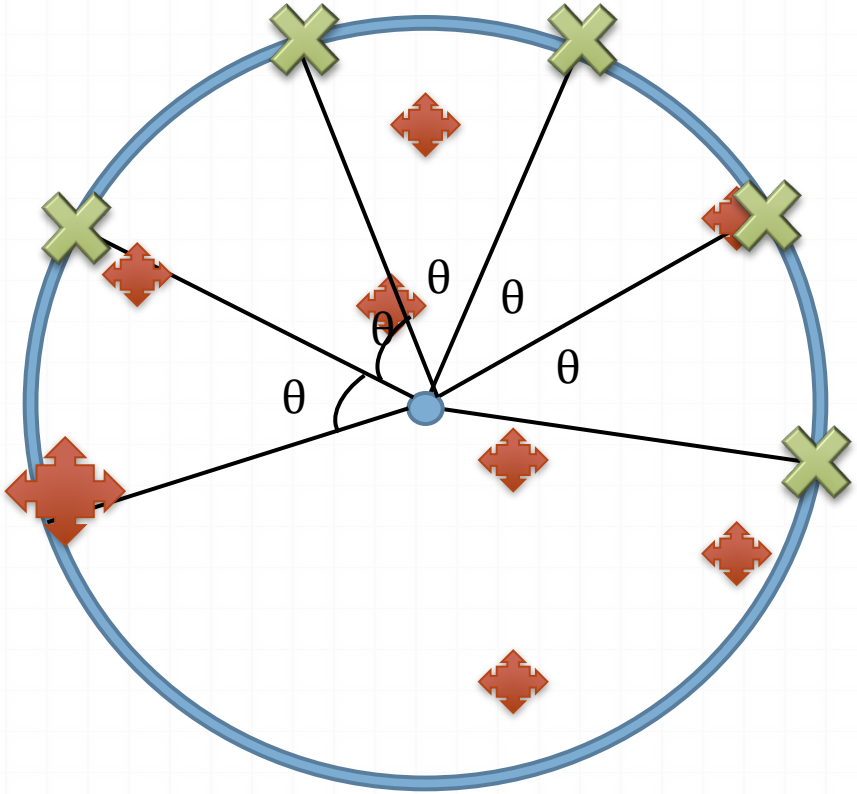
a[i].x = Center.x + x*cos(i*θ) – y*sin(i*θ)

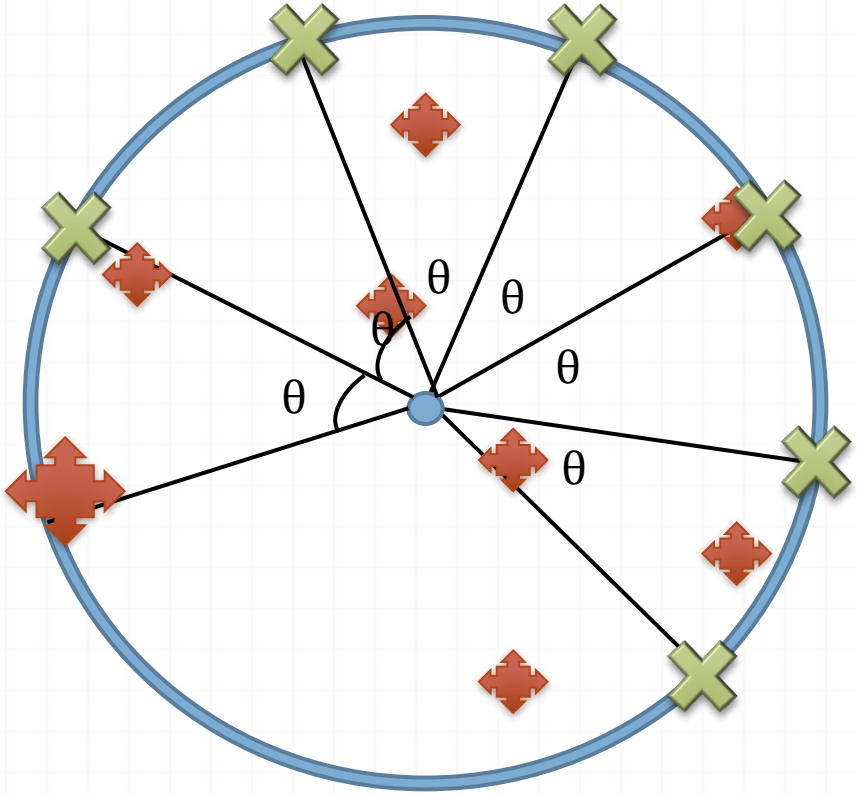
a[i].y = Center.y + y*cos(i*θ) – x*sin(i*θ)

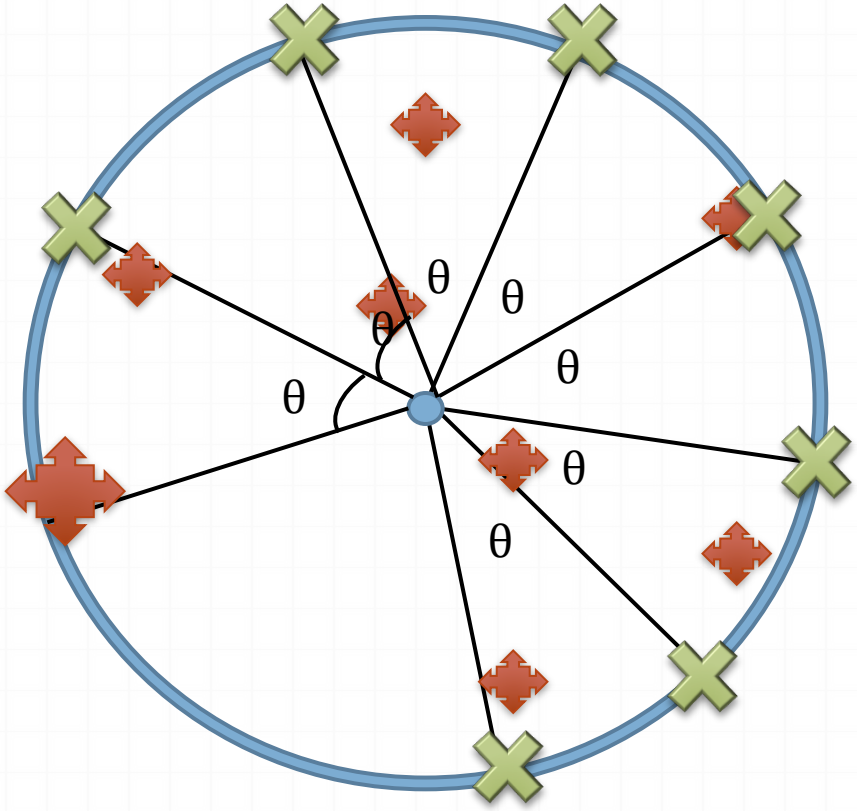

θ
θ

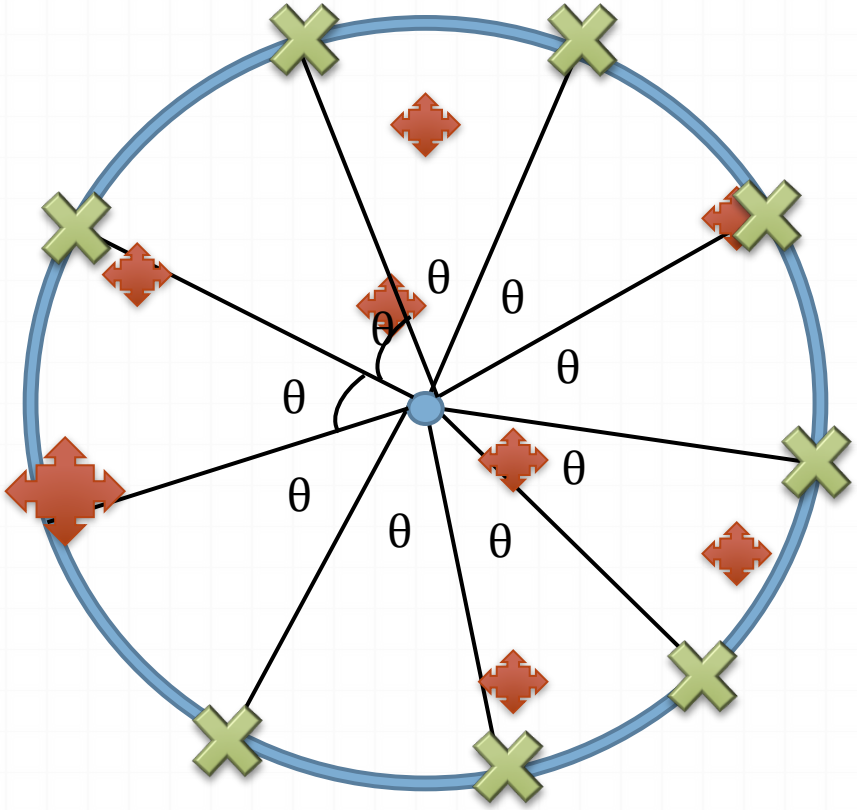

θ
θ
θ

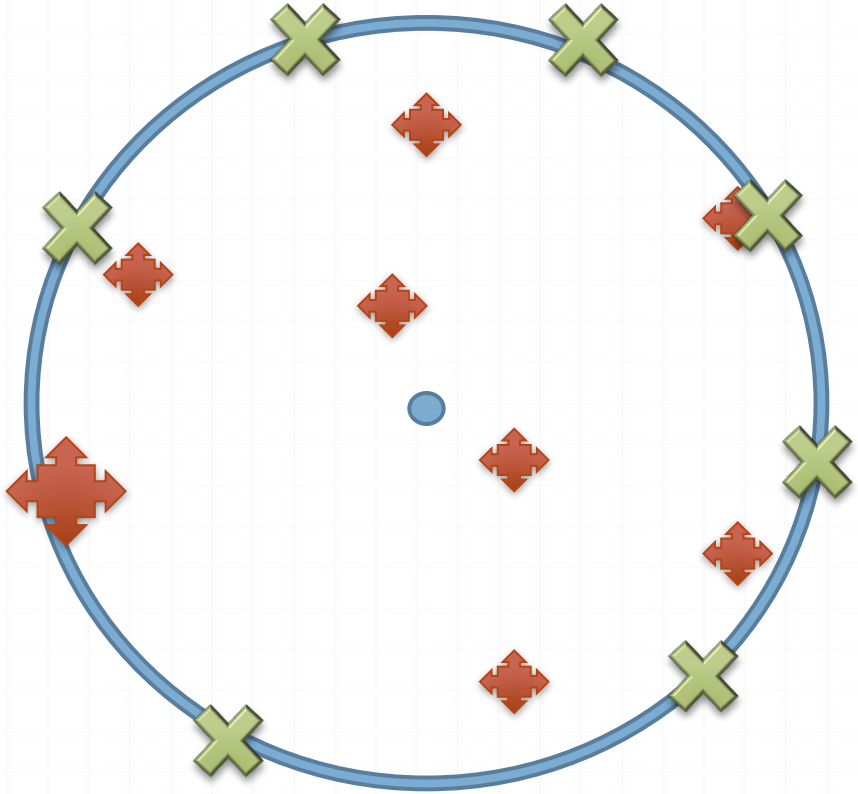

θ
θ
θ
θ


θ
θ
θ
θ
θ

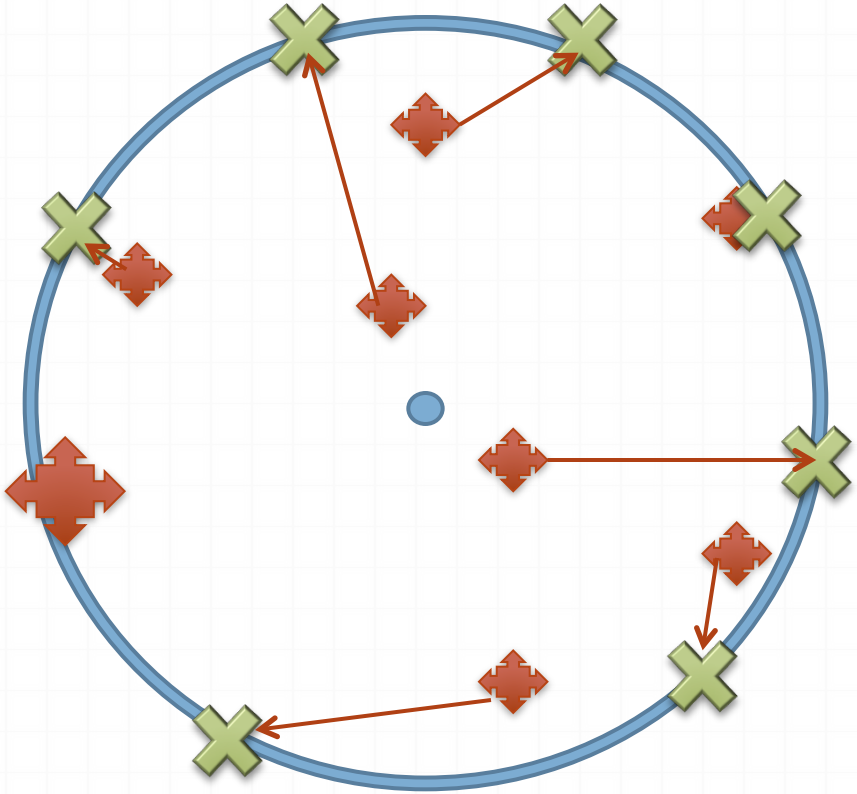

θ
θ
θ
θ
θ
θ


θ
θ
θ
θ
θ
θ
θ

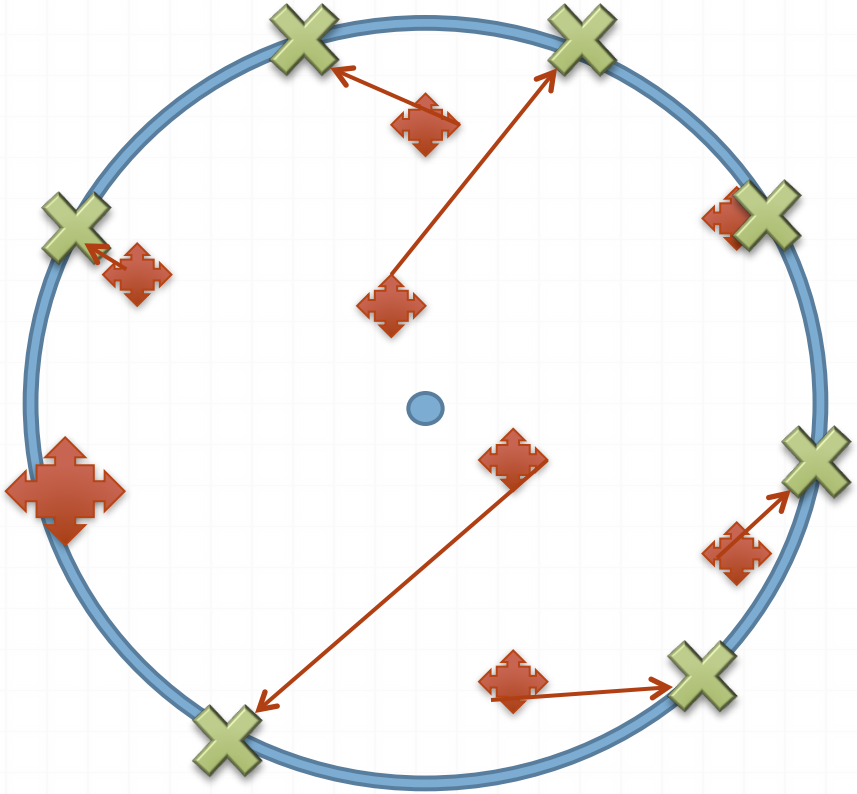

θ
θ
θ
θ
θ
θ
θ
θ



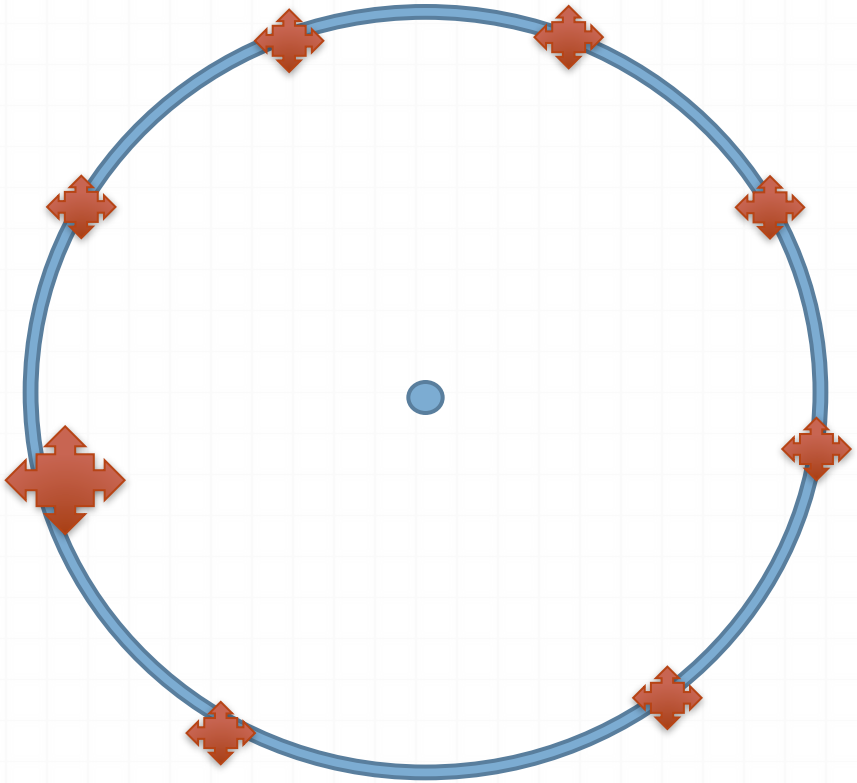
10) For each robot compute the closest cross



Other mapping?



11) Move robots to cross positions

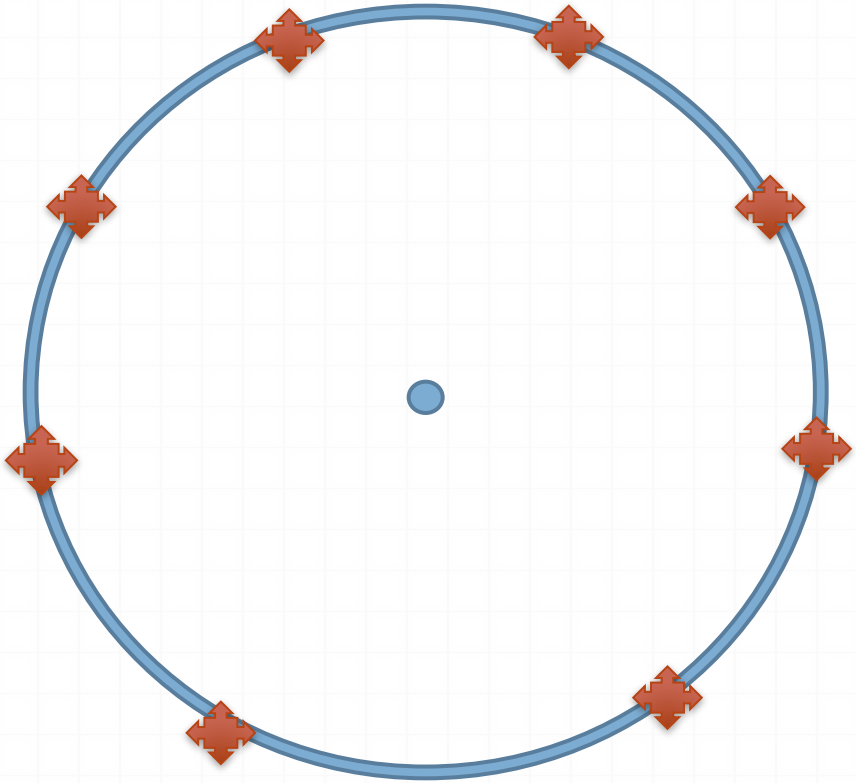


Algorithm *

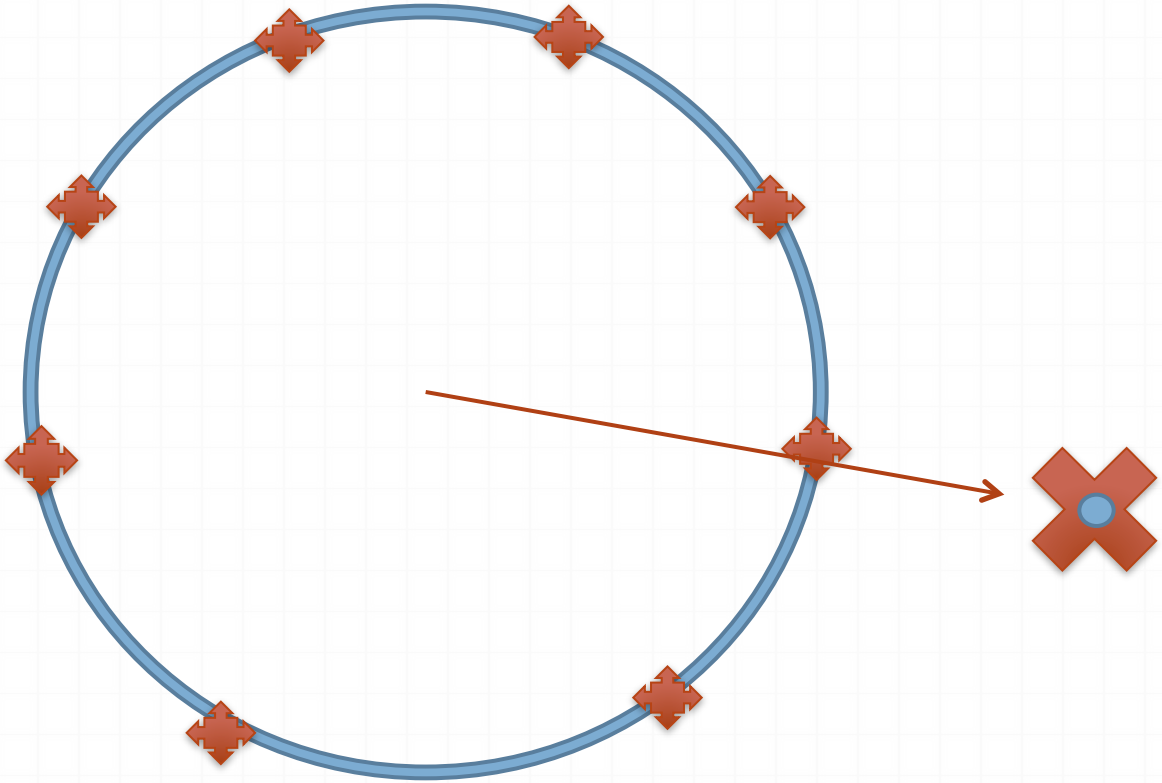
1. Smallest enclosing circle
2. Identify its center point
3. And radius
4. Make center point a «Leader» or a «Mass center»
5. Detect the robot on the enclosing circle
6. Count a number of robots
7. Count a needed angle between robots
8. Compute relative position
9. Compute position of all cross points
10. Compute the closest cross for each robot
11. Move robots to cross positions

Moving

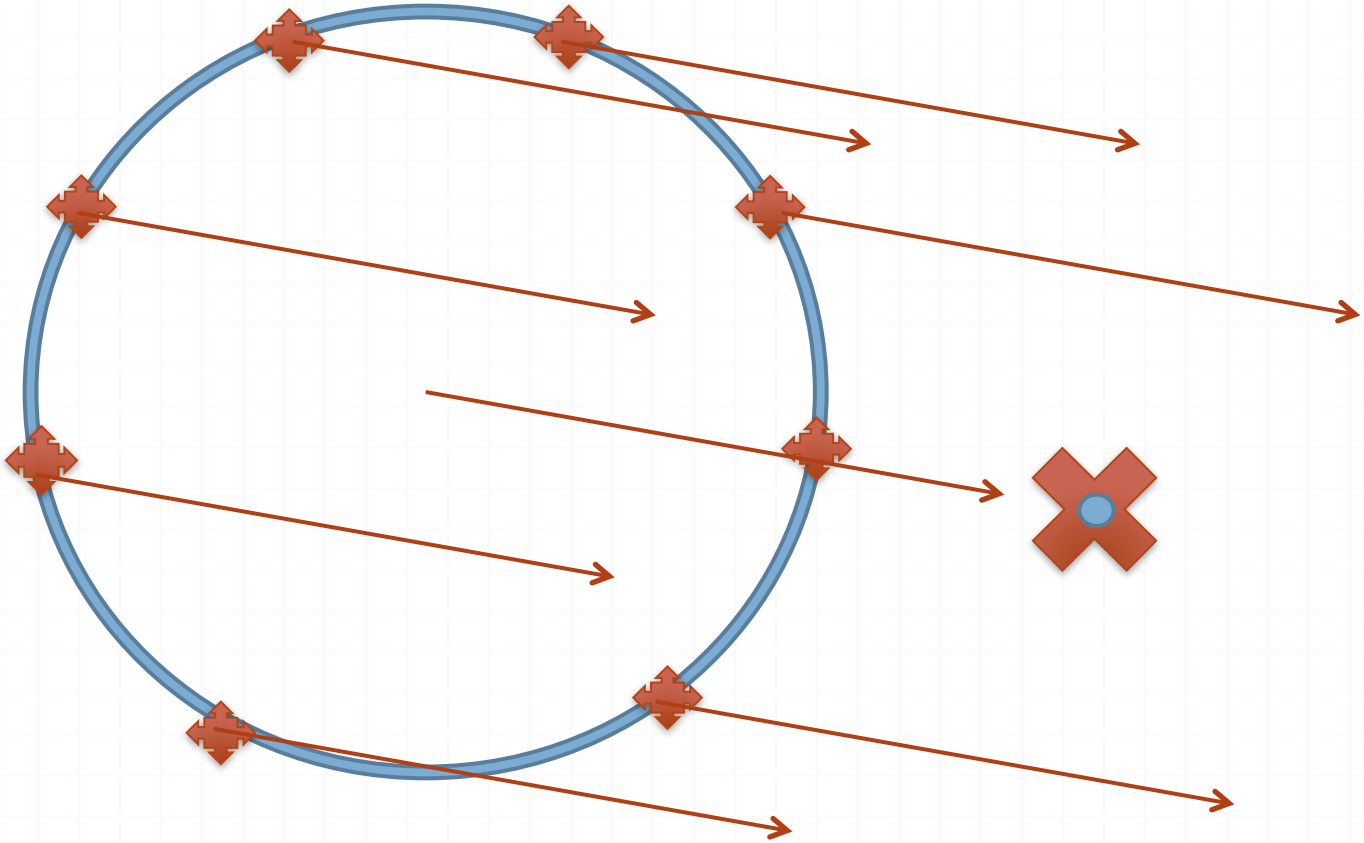
Identify the aim



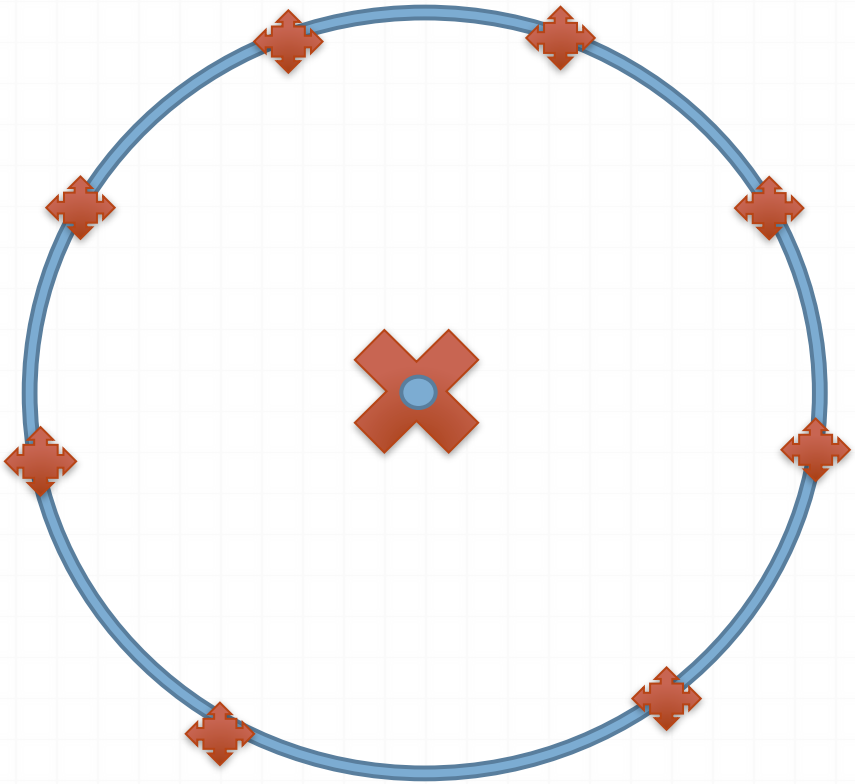
Compute a moving vector



Multiply each robot position with the same vector

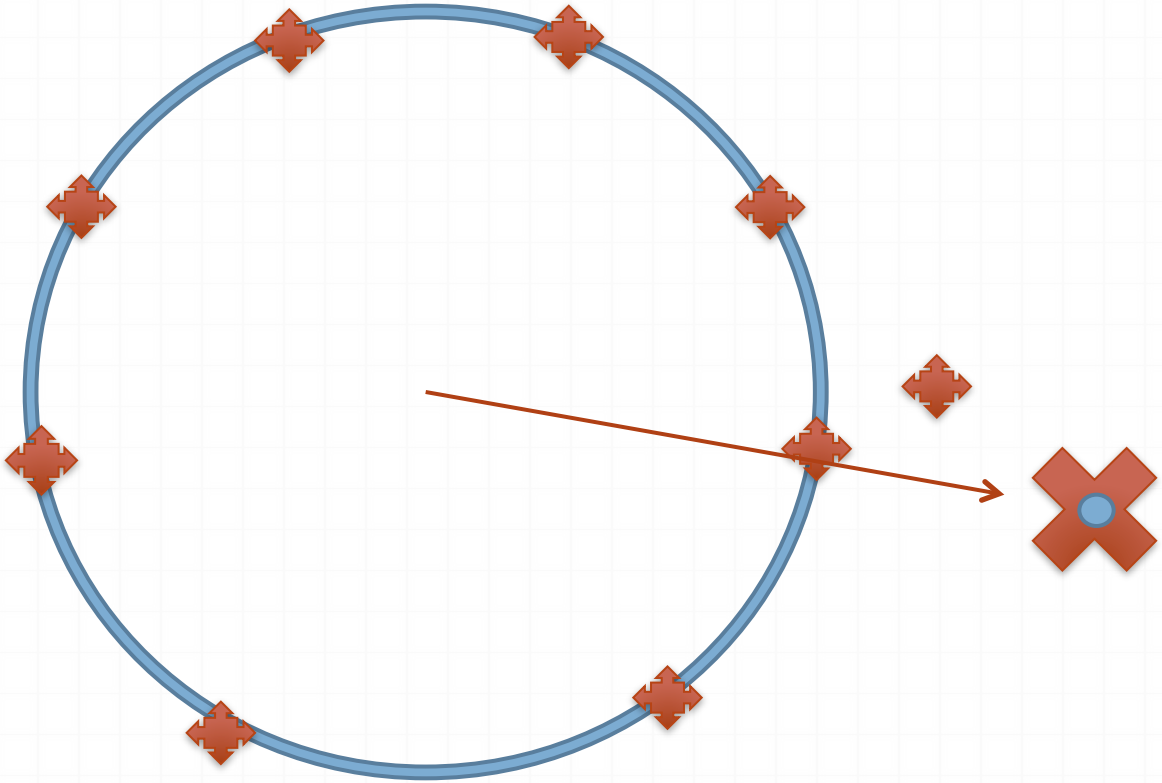


TADA!

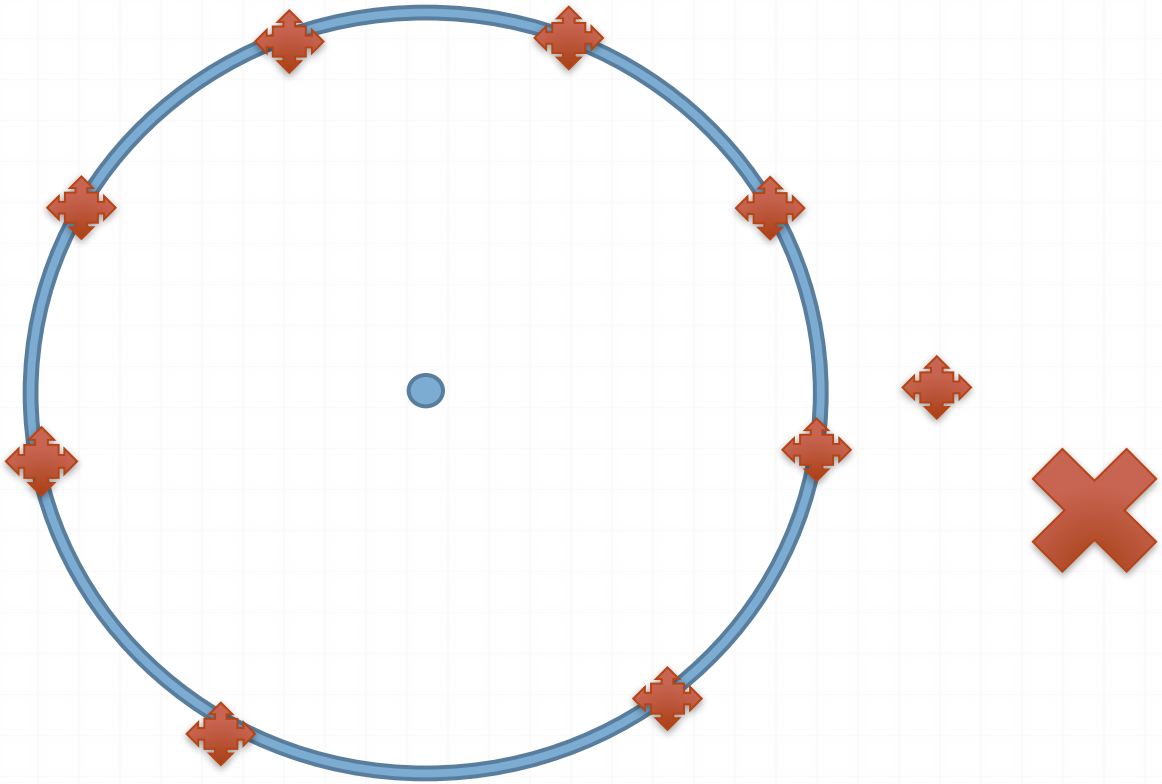


Joining of a new robot

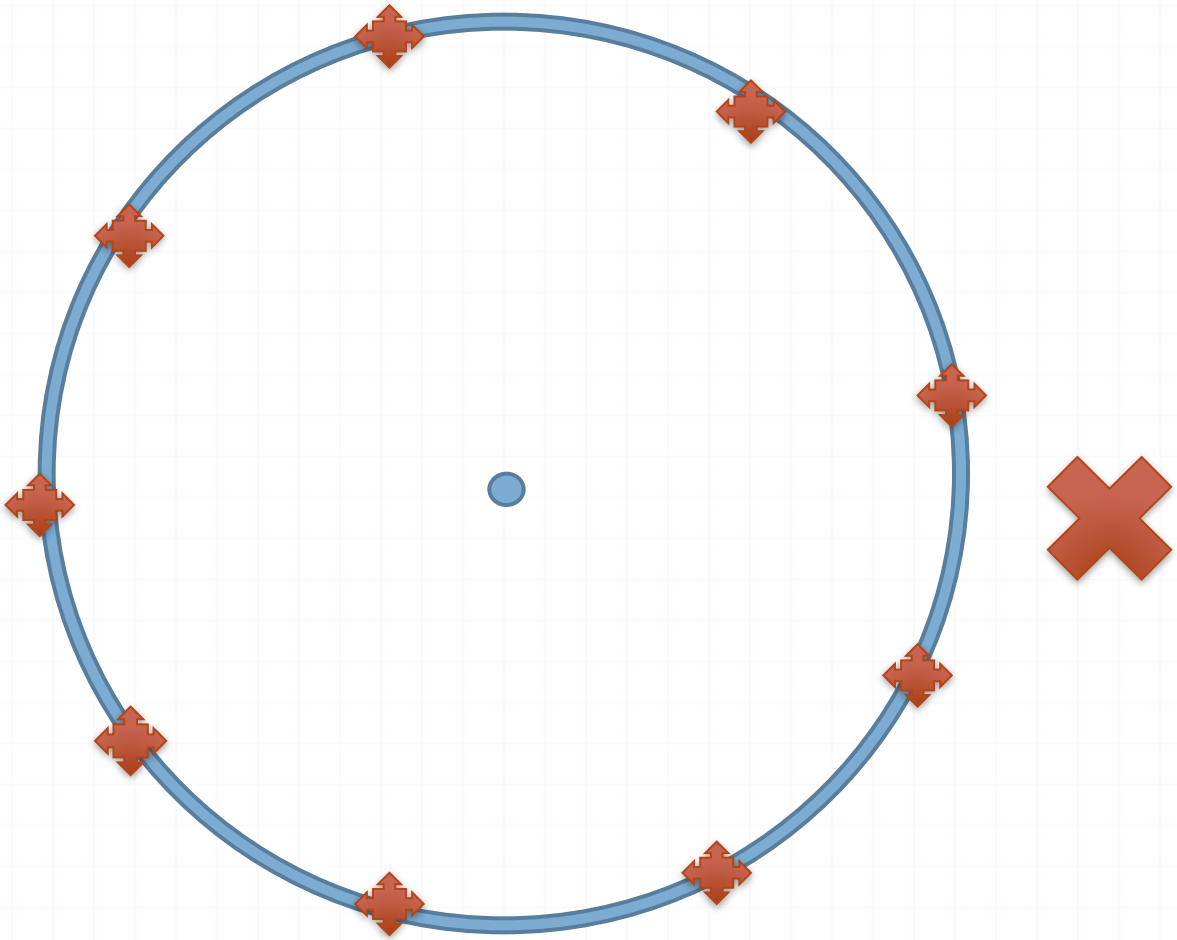
New robot



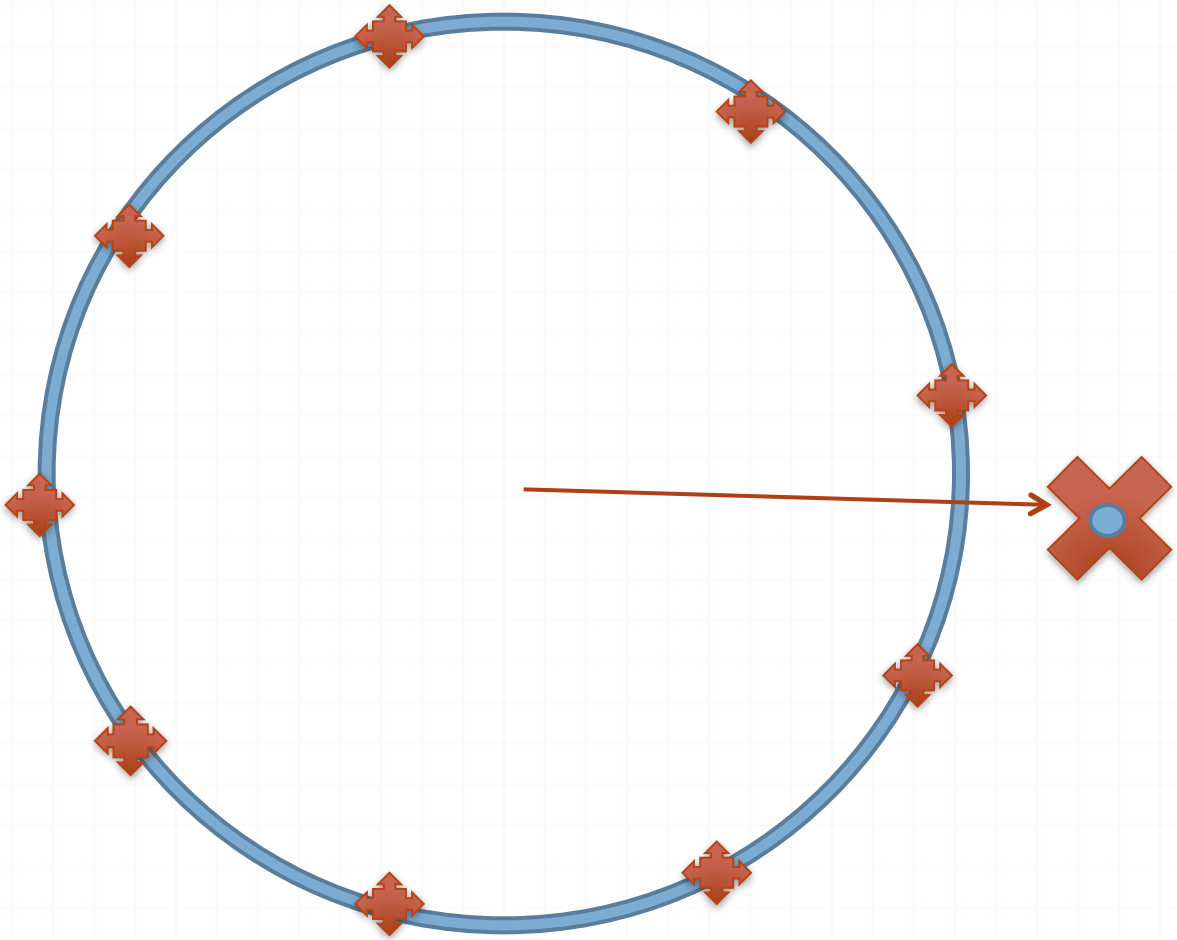
Stop



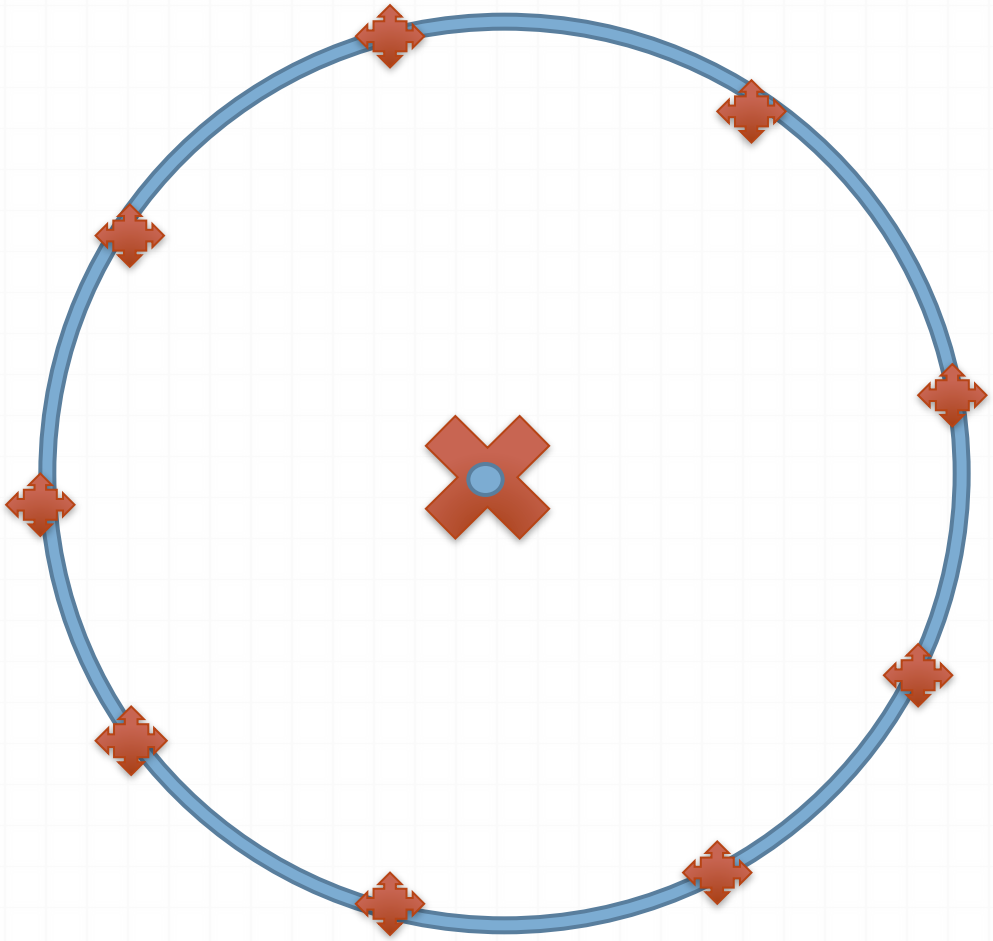
Reorganize circle due to Algorithm *



Continue moving



TADA!



Thank you for your attention