**Useful references and links**:

1. <https://robowiki.net/wiki/Radar>
   1. Has some tips and info about how the radars work.

**IMPORTANT**:

* Have a function set up for frequent scanning.
* This would rotate the gun with the radar.
* NOTE FOR LATER: Add a check velocity to methods to modify the search direction.

**Total targeting strategy behaviors**: 4

1. WeakRobot – the robot focuses on attacking a robot with low energy.
2. FocusAttack – once the robot targets a robot, it attacks it until it is destroyed.
3. Defense – the robot targets robots that attacked it.
4. NearestRobot **–** targets the closest robot.

**Additional genetic parameters**:

1. K
2. L

**Not Genetic, but targeting-related parameters**:

1. target – stores the name, energy level, and distance of the robot that our robot is trying to attack.
2. ticksSinceSeen – the number of ticks it has been since target has been detected.
3. searchDir – the direction, clockwise or counter-clockwise, to search for other robots.

**Targeting()**

Narrative: Conducts the general search for another robot.

Pre-condition: None

Post-condition: The robot has selected a target.

Turn the radar in the direction specified by searchDir unless interrupted or another robot is detected

Increment ticksSinceSeen

**OnScannedRobot(ScannedRobotEvent e)**

Narrative: The robot has detected another robot and will now select the appropriate behavior.

Pre-condition: None

Post-condition: The robot has performed its action on its target.

If targeting strategy is weak:

WeakRobot(e)

Else If targeting strategy is focused:

FocusedAttack(e)

Else If targeting strategy is defense:

Defense(e)

Else If targeting strategy is nearest:

NearestRobot(e)

Else

Report that there is an invalid targeting strategy

**WeakRobot(ScannedRobotEvent e)**

Narrative: The robot targets the weakest (lowest energy) robot it has encountered.

Pre-condition: None

Post-condition: The robot targets the weakest robot.

If target is null

Set target.name equal to e.name

Set target.energy equal to e.energy

Set target.distance equal to e.distance

If target.name is equal to e.name

Set target.energy equal to e.energy

Set ticksSinceSeen equal to zero

Else

If e.energy is less than target.energy

Set target.name equal to e.name

Set target.energy equal to e.energy

Set target.distance equal to e.distance

Set ticksSinceSeen equal to zero

Else

Return

Call BulletStrategy(e)

**FocusedAttack(ScannedRobotEvent e)**

Narrative: Once the robot has found another robot, it will target that robot until it is dead or cannot be found.

Pre-condition: None

Post-condition: Robot attacks its target.

If ticksSinceSeen is greater than or equal to 5

Set target to null

Set ticksSinceSeen equal to zero

If target is null

Set target.name equal to e.name

Set target.energy equal to e.energy

Set ticksSinceSeen equal to zero

If target.name is not equal to e.name

Return

Call BulletStrategy(e)

**Defense(ScannedRobotEvent e)**

Narrative: The robot targets any robot that attacked it.

Pre-condition: This method would require a HitByBullet Event to direct the scanner.

Post-condition: The robot will attack its attacker.

If target is null

Set target.name equal to e.name

Set target.energy equal to e.energy

**OnBulletHit(BulletHitEvent e)**

Narrative: Called whenever the robot is struck by a bullet.

Pre-condition: None

Post-condition:

**NearestRobot(ScannedRobotEvent e)**

Narrative: The robot targets whatever robot is closest to it.

Pre-condition: None

Post-condition:

If target is null

Set target.name equal to e.name

Set target.energy equal to e.energy

If target.name is equal to e.name

Set target.distance equal to e.distance

Else

If e.distance is less than target.distance

Set target.name equal to e.name

Set target.energy equal to e.energy

Set target.distance equal to e.distance

Else

Return

Call BulletStrategy(e)