RunAround Robocode

Team members:

Gabor Major

Mirella Glowinska

Daniel Vetrila

Robot Logic:

run main thread:

The gun is set to be independent of the robot body. This makes it more likely to find the enemy when the robot loses it as starts moving.

Robot scans for the enemy and the sentry. It starts the scan towards the centre of the map as the enemy is more likely to be there.

While loop:

This runs to reposition the robot based on the movement weights, by calling pickNewRandomPoint and turnAndMoveToPoint.

When the robot moves twice without the onScannedRobot getting called, then it stops and again scans towards the centre to find the enemy.

pickNewRandomPoint:

This is the main logic behind the operations of the robot. The robot picks a pseudorandom point to move to. The point can be North, South, East or West of the current robot position.

The weights for each direction is calculated as follows:

The closer the robot is to the wall the smaller the weight is for that direction.

And the closer the robot is to the sentry bot the smaller the weight is for that direction.

These two weights are then put in a 2:1 ratio, sentry:wall.

Robot

Robot

Sentry

turnAndMoveToPoint:

This takes in a point coordinates, turns the robot towards the point and moves to the target position.

scanTowardsCentre:

Gets the angle difference between the robot radar and the centre.

It then turns 360 depending on the angle returned, positive or negative.

onScannedRobot:

If the enemy is scanned then:

Information about the enemy is collected and its position is calculated.

Then the gun is turned towards the enemy position.

And if the robot gun heat is 0 then a shot of power of 3 is fired.

Else if the sentry is scanned:

The sentry’s position is calculated and saved for the movement algorithm.

onHitWall:

When the robot hits any of the walls, it turns towards the centre of the map and moves a third of the distance to it.

onStatus:

Collects all the information about the robot and saves them to variables to be used by the other parts of the code.

Utilities

getTurnAmount:

Returns the angle difference in degrees between the two angles input.

Returns between -180 and 180.

returnDegreesDifference:

Two coordinates and whatever we want to turn heading direction put in.

It calculates the degrees between the two coordinates and then calls getTurnAmount to get the required turn amount for the object we put in, body, gun, or radar.

convertToProperDegrees:

This just converts the Robocode direction angles to the sin, cos, tan system.

distanceToPoint:

Returns the distance between a point and robot position.