

Brian Norman (aka robotikdude)

Author of DeltaSquad

3/9/11

Most of my code is reused code from previous robots and I honestly don't remember a lot of it but I will do the best I can to answer your questions. I understand that this paper is rather bare-bones, but it was the best I could do under the time constraint and the fact that I am also currently taking classes.

1. Action performed on scanning an enemy robot

When a robot is scanned, the event is passed to a robot information manager (management.RobotManager). This class is responsible for keeping each enemy and friendly robot's information up to date. This information includes such things as x and y coordinates, heading, velocity, energy, time of latest scan, and some various delta values. For a more detailed look at what values are stored, look at the RobotData class under the package robot. This information is also sent to all the robot's teammates as well in the form of a custom message (communication.ScannedRobotMessage).

Another thing that occurs when a robot is scanned is that a table of VirtualGuns is updated if the robot is not already on this table. This process is managed by another manager class (managemenet.TargetingManager).

2. Action performed on hitting an enemy robot

No special action is performed when hitting an enemy robot.

3. Action performed on scanning a wall

When a robot scans an object, the point is stored and sent to teammates. As more and more points are added, lines can be drawn between these points forming a rough estimate. In later development versions, these lines were combined to form new lines which would increase run time when checking for intersections.

4. Action performed on hitting a wall

No special action is performed when hitting a wall.

5. Strategy to kill an enemy robot

To kill an enemy, my robots implore various targeting techniques. Each of these targeting methods is part of a system of virtual guns. The best virtual gun is chosen for each enemy by comparing a weight of the real and virtual hit ratios.

6. Strategy to capture a flag

The movement strategy for capturing a flag is the same as simply moving. I use a pretty generic version of minimum risk movement. When a flag is to be captured an attraction weight is placed at that point so the robot will move in that direction.

7. Strategy to defend one's own flag

The same strategy for capturing a flag is used for protecting our flag. A small weight is placed around the enemy's base so the robots will hang around there. When our flag is captured, the weight changes to be the flag so the robots will move to and attack the robot carrying the flag.

8. Action performed when flag is captured by enemy

See the previous question.

9. Any other techniques used by robots (for ex. Wave surfing or anti-gravity movements or others) should be explained clearly (implementation part only)

Most of my efforts were directed at an accurate representation of the objects on the battlefield. This was until, of course, homework happened and I was forced to put this aside.