Secrets from the Robocode masters: **Radar sweeps**

Eivind Bjarte Tjore

June , 2002

This tip was published in the "Cloak and turret: Learn secrets from the Robocode masters" article in the May 2002 issue of the IBM developerWorks journal.

One of the keys to success in Robocode is always having the most updated information on every enemy bot. To accomplish this, you must use your radar in such a way that it does not sweep more of the battlefield than necessary. For instance, when your robot is standing in a corner, it makes no sense to do a full 360-degree sweep.

This tip discusses the code that I am using in et.Predator to control the radar. It assumes that your robot is an AdvancedRobot and that you have an `Enemy` class (a class that holds the information from `ScannedRobotEvents`) and an `EnemyMap` class (a collection of Enemies). The `EnemyMap` that I am using extends `HashMap`, but you also could use `Hashtable` or another suitable class.

The `Enemy` class must have an `isUpdated()` function, which checks whether the time since the information last was updated (`getTime()` from the Robot minus `getTime()` from the `ScannedRobotEvent`) is below some limit. I currently use 16 as the limit.

In the initialization part of the robot (in `run()`, before the loop), use the following:

```
addCustomEvent(new
  RadarTurnCompleteCondition(this));
setAdjustRadarForGunTurn(true);
setTurnRadarRight(360);
```

Next, add the event handler for the custom event:

```
public void onCustomEvent(CustomEvent e) {
  if (e.getCondition() instanceof
    RadarTurnCompleteCondition) sweep();
}
```

You also should use `getScannedRobotEvents()` each round to ensure that all `ScannedRobotEvents` get recorded.

Finally, add the code in Listing 1 to control the radar sweeps. Note that the code uses the `int sign(double n)` method that returns the sign of the number n (-1 or 1). It also uses `normalRelativeAngle(double angle)` to normalize the input angle to an angle between -180 and 180 degrees.

Listing 1. Radar sweep method

```
private int radarDirection=1;

private void sweep() {
  double maxBearingAbs=0, maxBearing=0;
  int scannedBots=0;
  Iterator iterator = theEnemyMap.values().
    iterator();

  while(iterator.hasNext()) {
    Enemy tmp = (Enemy)iterator.next();

    if (tmp!=null && tmp.isUpdated()) {
      double bearing=normalRelativeAngle
        (getHeading() + tmp.getBearing()
         - getRadarHeading());
      if (Math.abs(bearing)>maxBearingAbs) {
        maxBearingAbs=Math.abs(bearing);
        maxBearing=bearing;
      }
      scannedBots++;
    }
  }

  double radarTurn=180*radarDirection;
  if (scannedBots==getOthers())
    radarTurn=maxBearing+sign(maxBearing)*22.5;

  setTurnRadarRight(radarTurn);
  radarDirection=sign(radarTurn);
}
```

The `sweep()` function determines the largest value for the absolute value of the bearing to the other robots. If all the robots are updated, it sets the radar to turn toward the enemy with the largest bearing from yours. If some robots have not been updated, it sets the radar to continue to sweep in the current direction.

Note that I add 22.5 degrees to the sweep, which is a safety margin because the robots have moved since the last scan. The motivation for choosing 22.5 as the constant is that the radar turns roughly 45 degrees per frame. By not using a constant, taking the distance to the robot with the largest bearing, and calculating the maximum value that the correct bearing can have, you can improve the algorithm.

Related topics

- Read all of the *Secrets from the Robocode masters*. This page will be updated as new tips become available.
- Robocode's creator, Mathew Nelson, maintains the official Robocode site. This should be the first stop for anyone serious about Robocode.
- "Rock 'em, sock 'em Robocode" (*developerWorks*, January 2002) disarms Robocode and starts you on your way to building your own customized lean, mean, fighting machine.
- In "Rock 'em, sock 'em Robocode: Round 2" (*developerWorks*, May 2002), Sing Li looks at advanced robot construction and team play.
- New to Java? Check out "Introduction to Java programming" (*developerWorks*, November 2004), a tutorial that steps you through the fundamentals of Java language programming.