

# *Robocode Basic Tactic for 1-vs-1*



# *Creating Robot*

*Create robot can be easy, Making your robot a  
winner is not!*



# *3 Main components*

**Radar – Controlling radar**

**Targeting - Attacking**

**Movement – Avoiding**



# *Radars*

Spinning Radar  
Radar Lock



# *Spinning Radar*

Simple spin of Radar, It's Ineffective

```
public void run() {  
    do {  
        turnRadarRight(360);  
    } while (true);  
}
```

# Radar Lock

This type of radar follows the robot around the battlefield.

```
public void run() {
    //---
    turnRadarRight(Double.POSITIVE_INFINITY);
    do {
        if(getRadarTurnRemaining() == 0.0)
            setTurnRadarRight(Double.POSITIVE_INFINITY);
    } while (true);
}

public void onScannedRobot(ScannedRobotEvent e) {
    double absoluteBearing = getHeading() + e.getBearing();
    double radarTurn = absoluteBearing - getRadarHeading();
    setTurnRadarRight(Utils.normalRelativeAngle(radarTurn));
}
```

# *Targeting*

Head-on Targeting

Linear Targeting

Circular Targeting

Random Targeting



# Head-on Targeting

Immediately fire when found the enemy

```
public void run() {  
    // - - -  
}  
public void onScannedRobot(ScannedRobotEvent e) {  
    fire(1);  
}
```

# *Linear Targeting*

A method of Targeting which assumes that the target will continue in the same direction at the same speed.



# *Circular Targeting*

Circular Targeting is used to hit bots that often move in circles or large arcs.

1. Measure the turnrate of your target bot by subtracting it's current heading with it's previous one.
  2. Use you turnrate to calculate the target's next position by apply to  $\sin()$  ,  $\cos()$  and enemy's velocity.
  3. Fire to the target's next position.
- 
-

# Random Targeting

Fire 3 power bullets randomly(or scatteringly) to the enemy between  $\pm 46.6$  degree from your radar heading when founded the enemy because it will not able to move an greater this angle.



# Random Targeting

**Why are we need to fire 3 power bullets?**

Because,

Velocity of bullet per tick =  $20 - 3 * \text{power} = 20 - 3 * 3 = 11$

Maximum velocity of tank per tick = 8

Maximum angle of target can move =  $\pm \arcsin(V_t/V_b)$   
 $= \pm \arcsin(8/11) = \pm 0.814 \text{ Radians} = \pm 46.6 \text{ Degrees}$

For other fire power bullets (1,2,4), need to recalculate degree.

# *Movement*

Pattern Movement  
Wavesurfing Movement



# *Pattern Movement*

LinearBots (ZigzagBots) – Moving in a straight line.  
Vulnerable to LinearTargeting.

CircleBots - Moves in circles. Vulnerable to  
CircularTargeting.

WallBots - Moves along the walls. Vulnerable to  
LinearTargeting, except when near corners.

RamBots - Moves to hit the enemy. Vulnerable to  
most targeting, but gets extra ram bonus.



# *Wavesurfing movement*

Moving with respect to where bullets are fired from.  
“Wavesurfing” was named because their movement is look like surfing(dodging) a wave of enemy's bullets.

Original Wavesurfing - It looks at the nearest approaching enemy bullet, tries to guess where it's going, and tries to be somewhere other than that.

---

---

# Wavesurfing movement(cont..)

Look-Ahead Wavesurfing  
– It looks at direction of set of approaching enemy bullets, tries to guess where it is safety side.



# *Spark your ideas*

Multi-style targeting/movement robot

Pattern Recognition

Historical Statistic

Traps

Tracking

Divide and Conquer

Neural Networks ^\_^

etc...

---

---

# *Conclusion*

Your works are ...

- (1) (Re)Design 3 Main Components
  - (2) (Re)Write it
  - (3) Test it
  - (4) Evaluate it
  - (5) Go back to (1)
- 
-

**Q&A**

Question and Answer



**Good luck in Robocode competition**

