นำเสนอที่ห้อง 502 ณ สถาบันเทคโนโลยีนานาชาติสิรินธร มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต วันที่ 27 ธันวาคม 2551

เรดาร์และการเคลื่อนที่

# พื้นฐานโรโบโค้ด

โดย กอบกฤตย์ วิริยะยุทธกร
kobkrit@gmail.com

# เรดาร์

- เรดาร์เป็นสิ่งที่จำเป็นมาก เพราะมันเป็นแหล่งรับข้อมูลศัตรูและสนามทั้งหมด
- ในหุ่นยนต์ที่มีการประมวลผลมากๆ จุดที่ดีที่สุดในการวางโค้ดคือคำสั่งแรกๆในเมดธ็อดรัน ซึ่งจะมั่นใจได้ว่า มันจะถูกประมวลทุกครั้ง ไม่ถูกข้ามไปเนื่องจากหมดเวลา

# อัลกอริทึมของเรดาร์แบบง่าย

- เรดาร์แบบหมุนไปเรื่อยๆ

```
public void run() {
    // หมุนเรดาร์ไปทางขวามากที่สุดเท่าที่จะมากได้
    turnRadarRightRadians(Double.POSITIVE_INFINITY);
}
```

- เรดาร์ล็อก(แบบง่าย)

```
public void run() {
    // หมุนเรดาร์ไปทางขวามากที่สุดเท่าที่จะมากได้
    turnRadarRightRadians(Double.POSITIVE_INFINITY);
}

public void onScannedRobot(ScannedRobotEvent e) {
    // เมื่อเจอหุ่นยนต์ ให้ปรับเรดาร์จากที่หมุนขวาตลอดเวลาให้หมุนองศากลับไปทางซ้ายเท่ากับองศาที่เหลือเมื่อเจอศัตรู
    setTurnRadarLeftRadians(getRadarTurnRemainingRadians());
}
```

# อัลกอริทึ่มของเรดาร์ล็อกแบบแคบ

```
import robocode.util.Utils;

 //...
 public void run() {
    // ...
    turnRadarRightRadians(Double.POSITIVE_INFINITY);
    do {
        // ค้นหาเป้าหมายใหม่
        scan();
    } while (true);
 }
 public void onScannedRobot(ScannedRobotEvent e) {
    //เซ็ตตัวแปร "องศาที่เรดาร์จะต้องหมุนเพิ่ม" ให้เท่ากับ
    double radarTurn =
        // องศาที่ตัวหุ่นยนต์ของเราหันอยู่เมื่อเทียบกับสนามรบ + องศาที่จุดค้นพบศัตรูเทียบกับองศาที่ตัวหุ่นยนต์เราหันอยู่
        getHeadingRadians() + e.getBearingRadians()
        // ลบด้วยองศาที่เรดาร์กำลังหันไปอยู่
        - getRadarHeadingRadians();
    //หมุนเรดาร์ไปทางขวาตามค่าของตัวแปร "องศาที่เรดาร์จะต้องหมุนเพิ่ม"
    setTurnRadarRightRadians(Utils.normalRelativeAngle(radarTurn));
    // ...
 }
```

# อัลกอริทึ่มของเรดาร์ล็อกแบบกว้าง

```
import robocode.util.Utils;

    public void run() {
      // ...
      while(true) {
        turnRadarRightRadians(Double.POSITIVE_INFINITY);
      }
    }

    public void onScannedRobot(ScannedRobotEvent e) {
      // เหมือนกันกับล็อกเป้าแบบแคบ
      double absoluteBearing = getHeadingRadians() + e.getBearingRadians();
      // Utils.normalRelativeAngle จะลบค่ามุมที่เกินกว่า 360 องศา ให้มีค่ามุมเดียวกันที่น้อยกว่า 360 องศา
    เช่น Utils.normalRelativeAngle(400) = 40 เป็นต้น จะได้ไม่เสียเวลาหมุนรอบตัวเองที่ไร้ค่าไปหนึ่งรอบ
      double radarTurn = Utils.normalRelativeAngle(absoluteBearing - getRadarHeadingRadians());

      // ความกว้างของหุ่นยนต์บวกด้วยสองเท่าของมุมเรดาร์ที่สามารถหมุนได้ใน 1 ตาเดิน ซึ่งมีค่าเท่ากับ36 pixels แต่ไม่เกินองศามากที่สุดที่หมุนได้ (45
    องศา = PI/4.0 , 1 PI = 180 องศา = 3.14 เรเดียน)
      double arcToScan = Math.min(Math.atan(36.0 / e.getDistance()), Math.PI/4.0);

      // เราต้องการส่งให้เรดาร์สแกนหาไปทางที่หุ่นยนต์ศัตรูเพิ่งพุ่งหนีไป
      radarTurn += (radarTurn < 0) ? -arcToScan : arcToScan;
      setTurnRadarRightRadians(radarTurn);
      // ...
    }
```

# การเคลื่อนที่แบบต่างๆ

# การเคลื่อนที่แบบต่างๆ

- การเคลื่อนที่เป็นเส้นตรง
- การเคลื่อนที่เป็นวงกลม
- การเคลื่อนที่เป็นซิกแซก
- การเคลื่อนที่เป็นมั่วซั่ว
- การเคลื่อนที่ขั้นสูง
  - การเคลื่อนที่แบบโต้ต้านแรงโน้มถ่วง
  - การเคลื่อนที่แบบโต้เคลื่อน

# การเคลื่อนที่แบบต่อต้านแรงโน้มถ่วง

- การเคลื่อนที่แบบต่อต้านแรงโน้มถ่วงเป็นการเคลื่อนที่ให้หุ่นยนต์ของตัวเองตีตัวออกห่างจากกลุ่มของศัตรูที่หนาแน่นให้มากที่สุด
- แต่เนื่องจากการแข่งขันแบบ 1-vs-1 ที่จะจัดใน Robocode Contest นั้นไม่มีกลุ่มของศัตรู(มีศัตรูแค่ตัวเดียว) การเคลื่อนที่แบบนี้จึงไม่เป็นผลเท่าไรนัก

# การเคลื่อนที่แบบโต้เคลื่อน

- การเคลื่อนที่แบบโต้เคลื่อนเป็นการเคลื่อนที่แบบตั้งฉากกับความเร่งของกระสุนแล้ว พยายามหลบด้วยเคลื่อนที่ไปทางซ้ายขวา
- พยายามเก็บข้อมูลทุกๆลูกกระสุนที่ยิงมาจากหุ่นยนต์ โดยเก็บไว้ในอาร์เรย์เก็บไว้
- ในทุกๆตาของหุ่นยนต์ เราจะเคลื่อนที่เพื่อหลบกระสุนไปทางที่ปลอดภัยที่สุด โดยคำนึงถึงอนาคตของกระสุนที่ยังไม่มาด้วย เลือกด้านที่ปลอดภัยที่สุด ไปทางซ้ายหรือขวา
  - ทิปของการเคลื่อนที่แบบโต้เคลื่อน
    - พยายามห่างจากหุ่นยนต์ศัตรูให้มากที่สุด
    - พยายามสังเกตเลือดของศัตรู เพราะสามารถบอกถึงประสิทธิภาพการหลบของเรา ถ้าศัตรูยิงเราโดนเยอะ เลือดศัตรูจะเพิ่ม แต่ถ้าเลือดของศัตรูลดลง แสดงว่าศัตรูยิงพลาดเยอะ