課題実験レポート

提出者　4 年情報工学科　11 番　小山内　穣

Java によるロボコードの作成

～strong sam～

目的

ロボコードを使い、実際にロボットを作成することにより Java 言語を学ぶ

開発環境

OS　　RedHat Linux9.0
言語　　Java 言語(J2SDK 1.4.2)

ロボコードとは

Java 言語を使って、作成したロボット戦車を対戦させるプログラム型ゲーム

ロボコードのルール

対戦形式　１対１以上(チーム戦もあり)
回数　　１ラウンドから 10 ラウンド
ロボット　ゲーム開始時 100 のエネルギーを持つ、エネルギーが０になった時点でロボットは破壊される
勝敗　　最後まで残ったロボットが勝ちではなく、多くの弾丸を当てるなど最終的に点数により勝敗をきめる

攻撃に関して

fire()と言う関数を使用する
つぎの式によりダメージを与える
　　パワー×4
更に２×（パワー‐1）のダメ‐ジを与える
同時に次式のエネルギ‐を得る
　　パワー×3

ロボット作成

関数

run()	これを記述することによって、ロボットの基本的な動作を設定する
fire()	弾丸を発射する関数
ahead()	ロボットを前進するための関数
turnLeft()	ロボットを回転するための関数
onHitByBullet	敵の弾丸が当たった時に呼び出される関数
onHitWall	壁に接触した時に呼び出される関数
onHitRobot	他のロボットに接触した時に呼び出される関数
onScanned Robot	ロボットを発見した時に呼び出される関数

作戦

・バトルが始まったら、壁に進んでいき壁に沿って移動して角に進んだところで，レーダーを回転させ敵ロボットを探す
・敵ロボットの弾丸が当たった場合、90 度方向を変えて移動、連続して弾丸が当たることを避ける
・敵ロボットと衝突した場合も、90 度方向を変えて移動する

実戦結果

・敵ロボットとの衝突により角度が変り動きがおかしくなることがあった。
・スタートする場所，他のロボットがいる場所により、結果がまったくちがう

まとめ

・スタート場所，他のロボットがいる場所により勝敗が変ってしまうというのは、作戦が不十分であったと考えられる
・今回のプログラムでは、相手の残りのエネルギーを考えて弾丸を発射するのではなく、敵がいなくなるまで撃ち続けるためエネルギーの無駄が生じてしまった

プログラムリスト

```
package Sam;
import rob code.*;
import java.awt.Color;
/**
* sam - a robot by (your name here)
*/
public class sam3 extends Robot
{
/**
* run: sam default behavior
*/
public void run() {
// After trying out your robot, try uncommenting the import at the top,
// and the next line:

setColors(new Color(10,20,20),Color.red,Color.yellow,Color.green);
//robotColor gunColor radarColor

turnLeft(getHeading()-90);
ahead(1000);
turnGunLeft(360);
turnLeft(90);

while(true){
ahead(1000);
```

```
turnGunLeft(360);

  }

}


//敵の弾丸が当たった時に起こるイベント
public void onHitByBullet(HitBulletEvent event){

        turnLeft(90); 90°回転
        ahead(1000);何かのイベントが起きるまで前進
        turnGunLeft(90); 90°砲台を回転

}


//壁にぶつかった時に起こるイベント
public void onHitWall(HitWallEvent event){

        turnLeft(90); //90°回転
        ahead(1000); //何かのイベントが起きるまで前進
        turnGunLeft(90)//90°砲台を回転

}


//敵ロボットを発見した時に起こるイベント
public void onScannedRobot(ScannedRobotEvent e) {

        fire(3);//パワー３で弾丸を発射
}
```

```
//敵ロボットと接触した時に起こるイベント
public void onHitRobot(HitRobotEvent event) {

turnLeft(90); //90°回転
ahead(1000); //1000 直進(壁にぶつかるまで直進)

}
```

考察

ロボコードについて

ロボコードにはいろいろな大会が開催されている

例えば、face2face、Robocode Rumble、Robocode ジャパンカップ、Survivalist League、Eternal Rumble がある。

- face2face

  1 対 1 のトーナメント戦

  各対戦では単純にスコアによって勝敗が決まるため、相手よりも多くのスコアを得る事が目的となる。
- Robocode Rumble 、Robocode ジャパンカップ

  これらの予選ではトーナメントではなく、リーグ戦によって順位が決まる。

  目的はスコアの合計を最大化すること、になります。

  必ずしも各対戦で相手よりも多くのスコアを出す必要はない。

  そのためトーナメントに比べて、攻撃的なロボットが有利になる。
- Survivalist League

  純粋に勝率によって順位を決めるため、生き残り率を重視したロボットが上位にランクされる。
- Eternal Rumble

  独自のレーティングを計算しているため、それに対応した攻撃性を考える必要があります。

感想

今回のロボコードの実験に関して、Java という言語を使用して作成していましたが、この言語は今までに使ったことが無かった言語であったにも関わらず、ロボット作成中に今までに使用したことの無い言語だから使いずらいなどの事はなかった。
ホームページを作ってレポートの提出とすることに関しても、今までホームページなど作ったことは無かったのですが、以外に簡単に作ることが出来た。ホームページを作ることは今回の課題実験の内容とは違うことなのかもしれないけれど、こちらの方も同時に勉強になったので良かったと思う。