

Guião do Trabalho Prático

Técnico de Informática e Sistemas

Fundamentos de Linguagem JAVA | UFCD: 0789

1. Introdução

No âmbito da UFCD 0789 – Fundamentos de Linguagem JAVA, construiu-se um robot utilizando a plataforma robocode, em linguagem Java que permite ao utilizador utilizando métodos, classes e instancias a criação de um robot virtual que será explicado nos pontos seguintes.

2. Movimento do Robot

O movimento do robot apresenta um movimento quase circular de 10 em 10 graus, continuo até ser atingido pelo adversário. Trata-se de um movimento com elevado desempenho porque consegue manter-se em combate com múltiplos robots sem nunca perder o ângulo de visão dos restantes robots. Este movimento circular só é parado quando o robot é atingido por uma bala.

3. Tiro do Robot

Em relação ao tiro do robot, pode-se dizer que este apresenta um tiro dinâmico compensado, porque não só realiza um conjunto de verificações com base na distância do adversário e no respectivo número de adversários mas também realiza a compensação de tiro com base no ângulo do adversário encontrado.

Em suma, o robot quando detecta um adversário e após verificar um conjunto de situações, realiza uma rotação pela direita até ficar de frente para o oponente e dispara segundo os critérios supracitados.

4. Comportamento no campo de batalha

Se o robô for atingido por uma bala, vira para a direita e procura o angulo do adversário, dispara tiro de intensidade 3, recua e faz scan.

Caso atinja uma parede, não pode falhar tiros. Recua e vira para a direita, vai em frente ou vira para a esquerda. Volta para traz e só pode bater uma vez na parede.

Se for atingido por outro robot vira a direita calcula o angulo do adversário, e dispara tiro de intensidade 3, seguindo em frente.

5. Extras

Foi implementada, a dança de vitória em que o robô faz rotações para a direita de, mudança na cor do robô nomeadamente vermelho, preto e branco.

6. Pontos Fortes vs. Pontos Francos

Ao bater no frente a frente 1 a 1, é um dos seus pontos mais fortes .

Um dos seus pontos mais fracos é quando fica retido nos cantos do ringue

7. Anexo (Código do Robot)

Exemplo:

```
package iefp;
import robocode.*;
import java.awt.Color;

// API help : http://robocode.sourceforge.net/docs/robocode/robocode/Robot.html

/**
 * DESTROYER - a robot by (Tiago Prates)
 */
public class DESTROYERRB extends Robot
{
    public int numtirosfalhados=0;
    public int batiparede=0;

    public void run() {
        setColors(Color.red,Color.black,Color.white);
        turnRadarRight(360);
        while(true) {
            // Replace the next 4 lines with any behavior you would lik

            int i;
            for(i=0;i<360;i=i+10){
                ahead(i);
                turnRight(i);
            }
            // turnRadarRight(20);
            scan();
        }
    }
}
```

```

}

/**
 * onScannedRobot: What to do when you see another robot
 */
public void onScannedRobot(ScannedRobotEvent e) {
    // Replace the next line with any behavior you would like

    //turnGunRight(e.getBearing());
    //turnGunRight(getHeading() - getGunHeading() + e.getBearing());
    //fire(3);
    //getHeading() --> graus da nossa direcção
    //getGunHeading() --> graus da nossa arma
    //e.getBearing() --> angulo do nosso adversario
    //em relação ao nosso robot

    double distancia;
    distancia=e.getDistance();
    turnRight(getHeading() - getGunHeading() + e.getBearing());
    if(getOthers()==2 || ((distancia<=350 || e.getEnergy()<50 || getEnergy()>50 ||
numtirosfalhados<2) && batiparede==0)){
        fire(3);
    }
    else if ((distancia>350 || getEnergy()<20 || numtirosfalhados<4)&&
batiparede==0){
        fire(2);
    }
    else{
        fire(3);
    }
    numtirosfalhados=numtirosfalhados+1;
    scan();
    batiparede=0;
}

/**
 * onHitByBullet: What to do when you're hit by a bullet
 */
public void onHitByBullet(HitByBulletEvent e) {
    // Replace the next line with any behavior you would like
    //back(10);

    //}

    turnRight(e.getBearing());
    turnRight(90);
    //fire(3);
    back(150);
    scan();
}

/**
 * onHitWall: What to do when you hit a wall

```

```
*/
public void onBulletHit(BulletHitEvent e) {
    // Replace the next line with any behavior you would like

    numtirosfalhados=0;

}

public void onHitWall(HitWallEvent e) {
    // Replace the next line with any behavior you would like
    //back(500);
    //turnRight(50);
    //ahead(100);
//}

    turnLeft(45);
    back(100);
    batiparede=1;
}

public void onHitRobot(HitRobotEvent e) {
    turnRight(e.getBearing());
    fire(3);
    ahead(100);
}

    public void onWin(WinEvent e) {
        // Replace the next line with any behavior you would like

        turnRight(36000);

        //onwin --> apos vitoria

    }

}

//FGDSFGFDGDF
```

TIAGO_PRATES