JULIA PAULINO ALABI julia.alabi@etec.sp.gov.br

MARIO ARMANDO TAKAHASHI MARTINS DIAS mario.dias2@etec.sp.gov.br

VICTOR GABRIEL FERREIRA THEODORO victor.theodoro@etec.sp.gov.br



**Os\_FakeNatty**

**BarbieScript**

ETEC REGISTRO

2023

**Resumo**

**Trecho 1 do código:**

**public class BarbieScript extends AdvancedRobot**

**{**

**int moveDirection=1;**

**public void run() {**

**setAdjustRadarForRobotTurn(true);**

**setBodyColor(Color.magenta);**

**setGunColor(Color.white);**

**setRadarColor(Color.magenta);**

**setScanColor(Color.white);**

**setBulletColor(Color.magenta);**

**setAdjustGunForRobotTurn(true);**

**turnRadarRightRadians(Double.POSITIVE\_INFINITY);**

**}**

**Explicação do Trecho 1 do código:**

* Estabelecendo a Herança entre BarbieScript da AdvancedRobot.
* Definindo a direção de movimento do robô.
* Mantém o radar parado enquanto o robô estiver se movendo.
* Definindo a cor do corpo do robô, arma, radar, scanner e as balas.
* Mantém a arma parada enquanto o robô estiver se movendo.
* Mantém o radar virado para a direita com base na medida de ângulo Radians.

**Trecho 2 do código:**

**public void onScannedRobot(ScannedRobotEvent e) {**

**double absBearing=e.getBearingRadians()+getHeadingRadians();//posição do inimigo**

**double latVel=e.getVelocity() \* Math.sin(e.getHeadingRadians() -absBearing);//velocidade do inimigo**

**double gunTurnAmt;//amount to turn our gun**

**setTurnRadarLeftRadians(getRadarTurnRemainingRadians());//lock on the radar**

**if(Math.random()>.9){**

**setMaxVelocity((12\*Math.random())+12);//muda a velocidade de forma aleatória**

**}**

**if (e.getDistance() > 150) {**

**gunTurnAmt = robocode.util.Utils.normalRelativeAngle(absBearing- getGunHeadingRadians()+latVel/22);//mudar angulo da arma**

**setTurnGunRightRadians(gunTurnAmt); //virar a arma**

**setTurnRightRadians(robocode.util.Utils.normalRelativeAngle(absBearing-getHeadingRadians()+latVel/getVelocity()));//vai até a posição que o inimigo inimigo supostamente vai estar**

**setAhead((e.getDistance() - 140)\*moveDirection);//ir para frente**

**setFire(3);//atirar**

**}**

**else{//se estiver mais perto**

**gunTurnAmt = robocode.util.Utils.normalRelativeAngle(absBearing- getGunHeadingRadians()+latVel/15);**

**setTurnGunRightRadians(gunTurnAmt);**

**setTurnLeft(-90-e.getBearing());**

**setAhead((e.getDistance() - 140)\*moveDirection);**

**setFire(3);**

**}**

**}**

**Explicação do Trecho 2 do código:**

* Método de Scan é chamado assim que o robô identifica um inimigo.
* Verificando a posição do inimigo.
* Verificando a velocidade do inimigo.
* Trava o radar no inimigo localizado a partir da medida de Ângulo.
* Permite a alteração da velocidade do robô de forma aleatória, pois desse forma fica mais difícil o inimigo prever o seu movimento.
* (Caso a distância seja maior que 150px, o robô muda ângulo da arma/move a posição da arma e segue o inimigo até sua posição e atira na potência 3).
* (Caso a distância do robô e do inimigo seja menor que 150px, o robô vai aumentar a velocidade e tentar prever onde o inimigo está, utilizando a medida Radians em ângulo para ver quanto a mira precisa se mover para chegar ao inimigo e atirar com a força 3.)

**Trecho 3 do código:**

**public void onHitWall(HitWallEvent e){**

**moveDirection=-moveDirection;//muda a direção quando bater na parede**

**}**

**/\*\***

**\*/**

**Explicação do Trecho 3 do código:**

* Método HitWall é chamado quando o robô atinge a parede.
* MoveDirection (1) controla a direção do robô, quando está em negativo (-1) a movimentação do robô será de ir para trás.

**Trecho 4 do código:**

**public void onWin(WinEvent e) {**

**for (int i = 0; i < 50; i++) {**

**turnRight(30);**

**turnLeft(30);**

**}**

**}**

**}**

**Explicação do Trecho 4 do código:**

* Método onWin é chamado quando o robô ganha a batalha.
* A estrutura de repetição em looping vai fazer o robô girar 30 graus para direita e 30 graus para esquerda, simulando um dança.