```java
package Tarefa;

import Auxiliar.ValoresSensores;

public class PassearFugirAgressivo extends TarefaRobot {

    private int velocidadeEsquerda;
    private int velocidadeDireita;
    private int lastVelocidadeEsquerda;
    private int lastVelocidadeDireita;
    private int counter;
    private int[] speeds;

    private static final int obstaculoLimiar = 14;
    private static final int valorBase = 4;
    private static final int modificador = 2;
    private static final int limiarRepeticao = 1;

    public PassearFugirAgressivo(HemissonRobot robot, AvaliadorAmbiente aval) {
        super(robot,aval);
        this.lastVelocidadeDireita = 0;
        this.lastVelocidadeEsquerda = 0;
        this.counter = 0;
        this.speeds = new int[2];
    }

    public void runOnce() {
        valoresSensores = robot.readSonar();
        velocidadeEsquerda = valorBase +
 obstaculo(valoresSensores[ValoresSensores.esquerda]) +
 obstaculo(valoresSensores[ValoresSensores.frenteEsquerda]) -
 obstaculo(valoresSensores[ValoresSensores.direita]) -
 obstaculo(valoresSensores[ValoresSensores.frenteDireita]);
        velocidadeDireita = valorBase +
 obstaculo(valoresSensores[ValoresSensores.direita]) +
 obstaculo(valoresSensores[ValoresSensores.frenteDireita]) -
 obstaculo(valoresSensores[ValoresSensores.esquerda]) -
 obstaculo(valoresSensores[ValoresSensores.frenteEsquerda]);
        if(!(velocidadeEsquerda == valorBase + 1 && velocidadeDireita == valorBase + 1)
 && !(velocidadeEsquerda == valorBase + 2 && velocidadeDireita == valorBase + 2)) {
            if(obstaculo(valoresSensores[ValoresSensores.frenteEsquerda]) != 0 &&
 obstaculo(valoresSensores[ValoresSensores.frenteDireita]) != 0) {
                robot.beeper(1);
                speeds[0] = -5 + random();
                speeds[1] = -5 + random();
            } else if(obstaculo(valoresSensores[ValoresSensores.esquerda]) != 0 ||
 obstaculo(valoresSensores[ValoresSensores.frenteEsquerda]) != 0 ||
 obstaculo(valoresSensores[ValoresSensores.direita]) != 0 ||
 obstaculo(valoresSensores[ValoresSensores.frenteDireita]) != 0) {
                robot.beeper(2);
                checkVelocidades(velocidadeEsquerda,velocidadeDireita);
            } else
                introduceRandomChange(velocidadeEsquerda,velocidadeDireita);

            robot.setSpeed(speeds[0],speeds[1]);
            System.err.println("Setting speed: " + velocidadeEsquerda + ":" +
 velocidadeDireita);
        } else {
            robot.setSpeed(-5,-5);
            robot.setSpeed(random(),random());
        }
    }

    private void introduceRandomChange(int velocidadeEsquerda, int velocidadeDireita) {
        speeds[0] = velocidadeEsquerda + random();
        speeds[1] = velocidadeDireita + random();
```

```
    }

    private int random() {
        int sentido = -1 * rand.nextInt(2);
        if(sentido != -1)
            sentido = 1;

        return sentido * rand.nextInt(6);
    }

    private void checkVelocidades(int velocidadeEsquerda, int velocidadeDireita) {
        speeds[0] = velocidadeEsquerda;
        speeds[1] = velocidadeDireita;

        if(velocidadeEsquerda == lastVelocidadeEsquerda && velocidadeDireita ==
 lastVelocidadeDireita)
        {
            counter ++;
            if(counter > limiarRepeticao) {
                counter = 0;
                speeds[0] = -1 * velocidadeDireita;
                speeds[1] = -1 * velocidadeEsquerda;
            }
        } else {
            counter = 0;
            lastVelocidadeEsquerda = velocidadeEsquerda;
            lastVelocidadeDireita = velocidadeDireita;
        }
    }

    private int obstaculo(int valor) {
        if (valor >= obstaculoLimiar)
            return modificador;
        else
            return 0;
    }
}
```