

---

**Administração Central**

# ROBOCÓDE

ROBÓTICA PAULA SOUZA



**2019**

**São Paulo**

**Administração Central**

**Material Didático sobre Robocode**

## 1 Anatomia do Tanque

Neste material vamos aprender alguns métodos da classe **Robot** para movimentação do canhão (Gun) e do radar (Radar).

No material **Conhecendo o Robô**, vocês já aprenderam um pouco sobre a anatomia do robô, que é formado pelo veículo, canhão e radar. Esses objetos podem ser controlados separadamente através de métodos específicos.

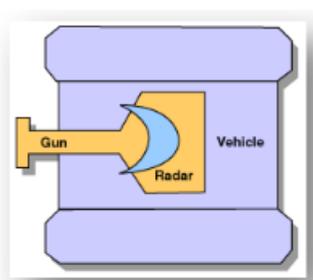


Figura 1 – Anatomia do Robô  
Fonte: do autor

## 2 Métodos para o canhão

O canhão (Gun) é o responsável pela realização dos disparos de balas (bullets) do nosso tanque de guerra e pode ser girado para a esquerda ou para a direita.

**fire()** - Dispara uma bala. O intervalo válido de energia é de 0,1 a 3. A bala vai viajar na direção em que o canhão está apontando. A bala fará (4 \* poder) dano se bater em outro robô. Se a potência for maior que 1, causará um dano adicional de 2 \* (potência - 1). Você terá (3 \* poder) de volta se você acertar o outro robô. Um evento será gerado quando a bala atingir um robô, uma parede ou outra bala. Esta chamada é executada imediatamente. O parâmetro passado neste método é o poder (power), que é a energia dada à bala e subtraída da sua energia.

---

### Administração Central

**turnGunLeft()** - Gira o canhão do seu robô para a esquerda. Essa chamada é executada imediatamente e não retorna até que seja concluída. O parâmetro passado para que o giro aconteça é realizado em até 360 graus. Exemplo: turnGunLeft(90) – esse método fará com que o canhão gire 90 graus para a esquerda.

**turnGunRight()** - Gira o canhão do seu robô para a direita. Essa chamada é executada imediatamente e não retorna até que seja concluída. O parâmetro passado para que o giro aconteça é realizado em até 360 graus. Exemplo: turnGunRight(90) – esse método fará com que o canhão gire 90 graus para a direita.

Exemplo de chamada de método dentro do nosso robô **Tanque.java** criando anteriormente:

```
package meustanques;
import robocode.*;
import java.awt.Color;
public class Tanque extends Robot
{
    public void run() {
        setColors(Color.YELLOW,Color.BLUE,Color.RED,Color.BLACK,Color.GREEN);
        while(true){
            turnGunLeft(90);
            ahead(100);
            back(500);
            turnLeft(45);
            turnRight(180);
            turnGunRight(45);
            fire(1);
        }
    }
}
```

### Administração Central

Se você observar bem nosso tanque, irá perceber que faz o giro do canhão separadamente do corpo do veículo. Porém nossa lógica está um pouco confusa, pois a bala está sendo disparada sem sabermos se há inimigo para que seja acertado neste intervalo.

### 3 Métodos para o radar

O radar permite detectar outros robôs presentes em um alcance de até 1200 pixels e pode ser rotacionado em um ângulo de, no máximo, 45 graus a cada turno. Usando o arco de escaneamento e aumentando o redimensionamento do campo de batalha é possível observar o limite de alcance do radar.

O redimensionamento do campo de batalha é feito quando configuramos a batalha. Podemos fazer alguns testes somente para efeito de aprendizado mudando o redimensionamento para 2000X2000, por exemplo. Mas, para efeito de competição, observando as regras para o torneio, esse dimensionamento deve ser de 800X600.

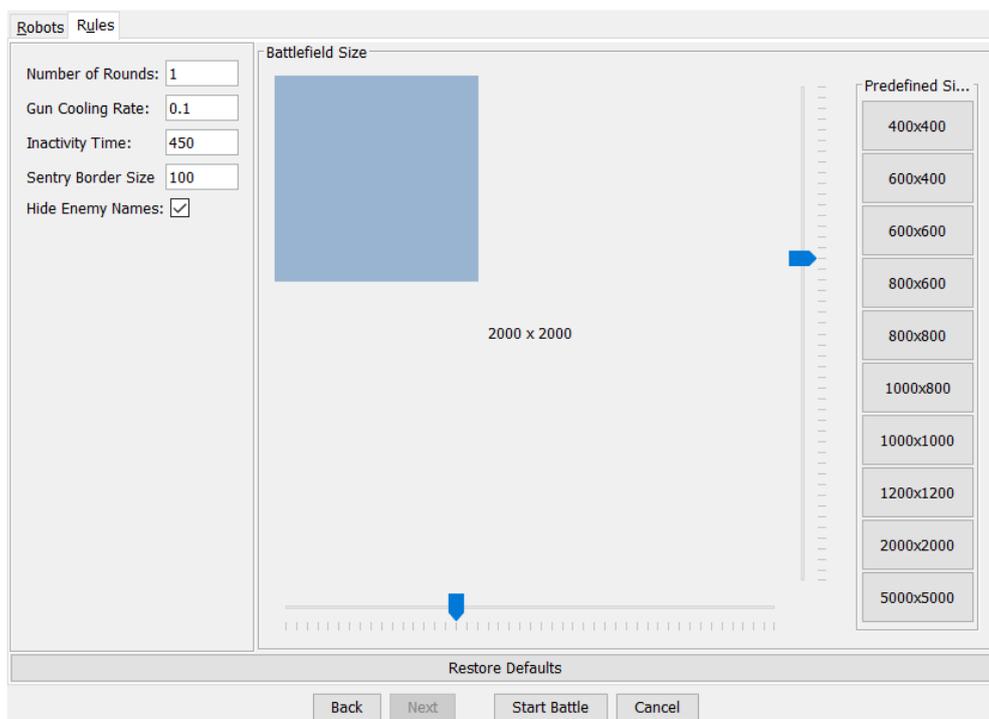


Figura 2 – Redimensionamento da arena de batalha

Fonte: do autor

## Administração Central

Para utilizar o arco de escaneamento precisamos habilitá-lo no Robocode. Esse procedimento é feito através do menu Options → Preferences.

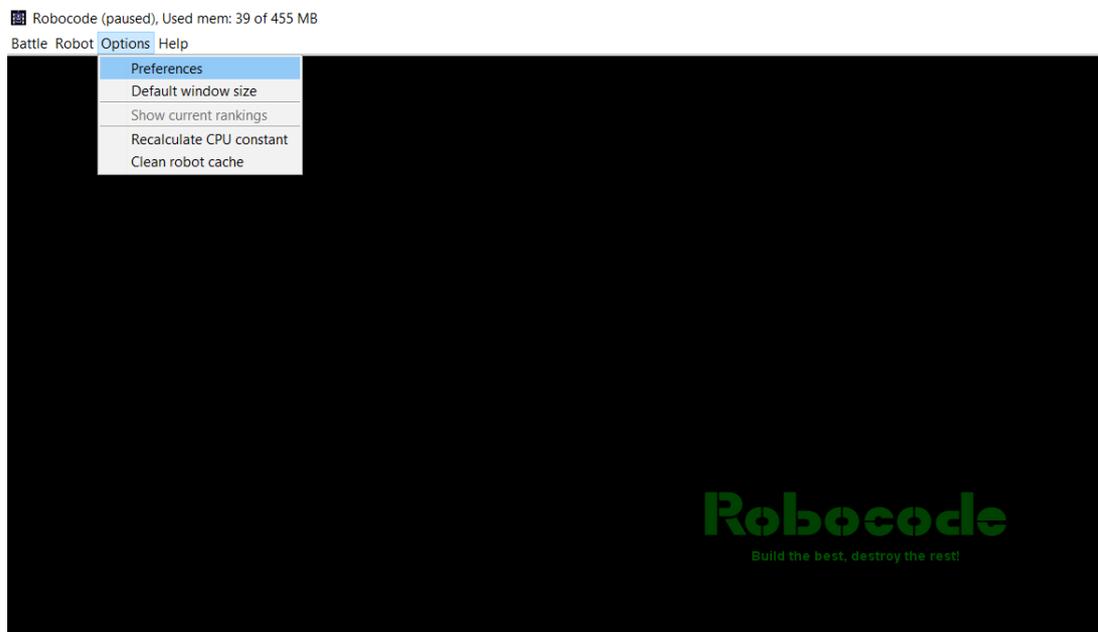


Figura 3 – Menu Options  
Fonte: do autor

Quando a janela Preferences for aberta, devemos selecionar a aba View Options e habilitar a opção Visible Scan Arcs.

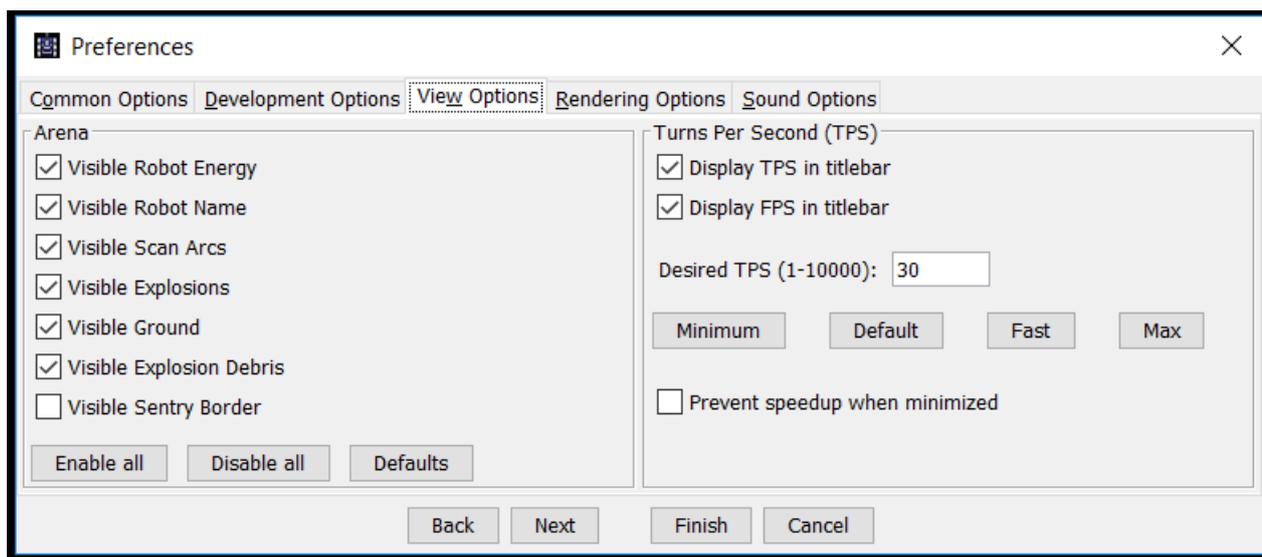


Figura 4 – Janela Preferences  
Fonte: do autor

## Administração Central

Em Preferences é possível habilitar outras opções, como por exemplo, deixar o nome dos robôs visíveis, a energia, as explosões, as opções de sons durante a batalha, etc.

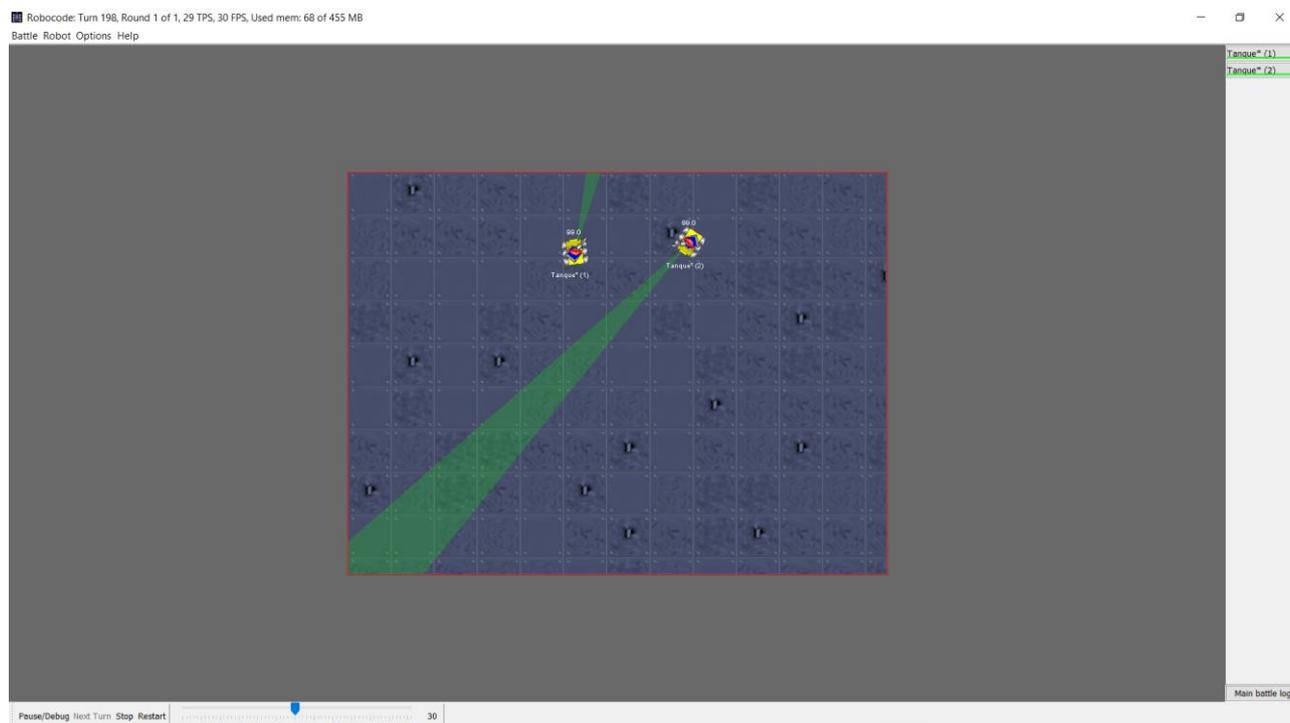


Figura 5 – Arena de batalha com Scan Arcs ativado  
Fonte: do autor

**turnRadarLeft()** - Gira o radar do seu robô para a esquerda. Essa chamada é executada imediatamente e não retorna até que seja concluída. O radar girará quando o canhão girar, e deverão estar na mesma direção. O parâmetro será passado em graus.

**turnRadarRight()** - Gira o radar do seu robô para a direita. Essa chamada é executada imediatamente e não retorna até que seja concluída. O radar girará quando o canhão girar, e deverão estar na mesma direção. O parâmetro será passado em graus.

## 4 Eventos utilizados

No Robocode temos alguns eventos específicos para trabalharmos com esses métodos vistos anteriormente e melhorarmos nossa lógica de programação e conseqüentemente nossa estratégia de jogo.

---

### Administração Central

**onScannedRobot()** – Esse evento é executado quando o radar do seu robô encontra um robô inimigo. Com ele é possível saber a energia, distância e ângulo do robô inimigo. Você deve substituí-lo em seu robô se quiser ser informado sobre esse evento. Este evento será chamado automaticamente se houver um robô no alcance de seu radar. O rolamento é relativo ao rumo do seu robô. O parâmetro passado será event, que é o evento definido no jogo, e trabalha com os métodos da classe ScannedRobotEvent.

Sintaxe: `public void onScannedRobot(ScannedRobotEvent event)`

### 5 Alguns métodos da classe ScannedRobotEvent

**getDistance()** - Retorna a distância do robô adversário em relação ao seu robô.

**getEnergy()** - Retorna a energia do robô adversário.

**getHeading()** - Retorna o ângulo em graus do adversário em relação a tela.

**getVelocity()** - retorna a velocidade do robô scaneado

### 6 Referências

Computer Science. **Class**. Disponível em:

<<http://www.cs.xu.edu/csci170/02s/robocode/javadoc/>>. Acesso em 20/06/2019.

Computer Science. **ScannedRobotEvent**. Disponível em:

<<http://www.cs.xu.edu/csci170/02s/robocode/javadoc/robocode/ScannedRobotEvent.html>>. Acesso em 20/07/2019.