

**Administração Central**

Unidade do Ensino Médio e Técnico – Cetec Capacitações



**ROBÓTICA**  
● ● ●  
**Paula Souza**

2019  
São Paulo

## Administração Central

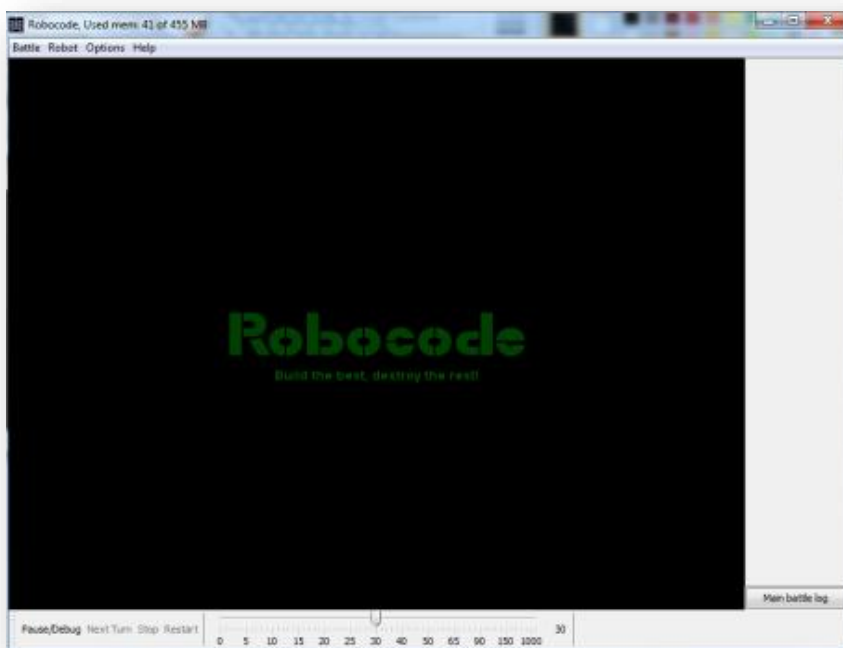
Unidade do Ensino Médio e Técnico – Cetec Capacitações

# Material Didático sobre Robocode Conhecendo o Robô

## 1 Criando nosso primeiro Robô

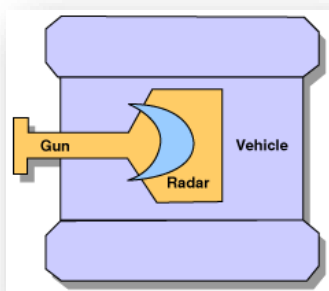
### 1.1 Ambiente de Batalha

O ambiente de batalha é o local onde será realizada a guerra entre os Robôs. Abaixo a imagem do ambiente:



### 1.2 Anatomia do Robô

O robô é um tanque que contém um canhão, um radar e o veículo, onde é possível programar estes três itens.

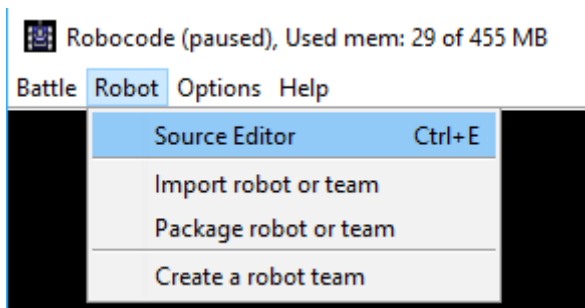


## Administração Central

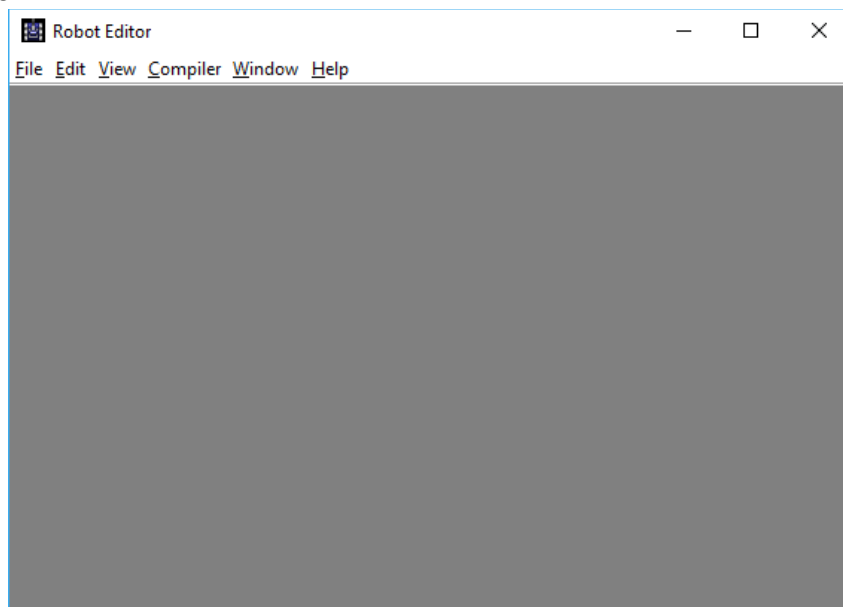
Unidade do Ensino Médio e Técnico – Cetec Capacitações

### 1.3 Criando nosso primeiro Robô

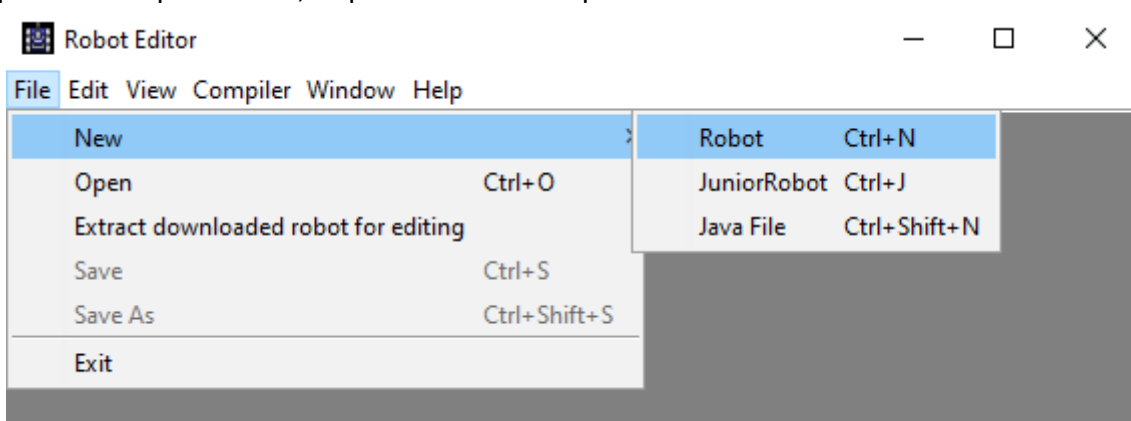
Para criar seu primeiro robô, no menu clique em **Robot** e depois em **Source Editor**:



Abrirá o seguinte editor:



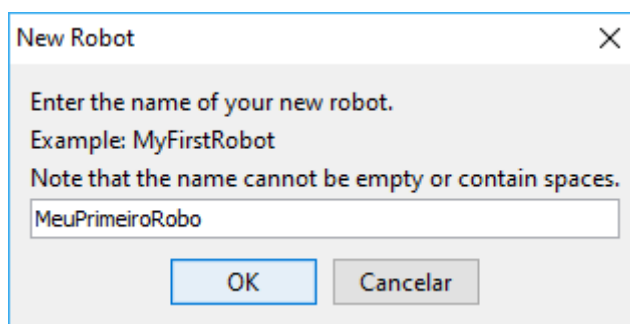
No Editor que abriu clique em **File**, depois em **New** e depois em **Robot**.



## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

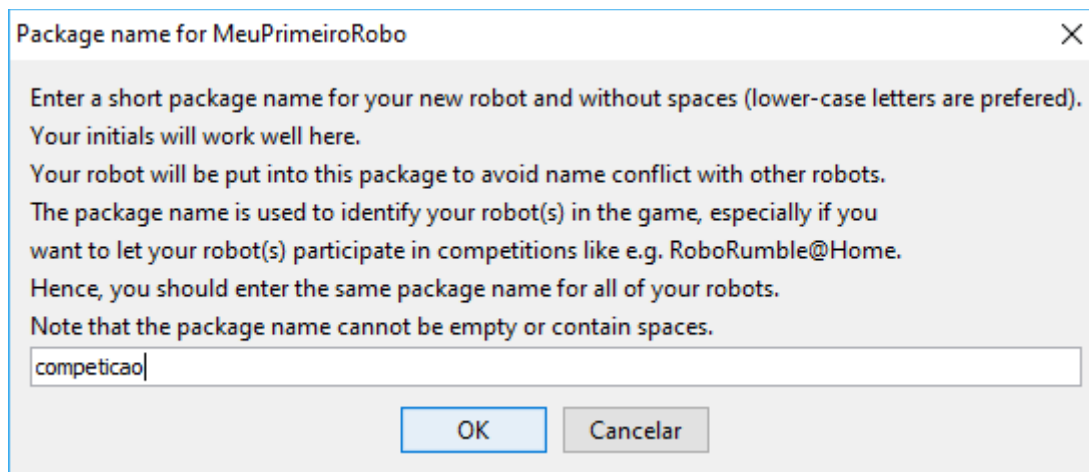
Como este é o nosso primeiro robô, estou nomeando de **MeuPrimeiroRobo** e clique em **OK**.  
(Obs: Sempre inicie o nome com a primeira letra em maiúsculo e sendo palavra composta as próximas letras iniciais também maiúsculas).



New Robot

Enter the name of your new robot.  
Example: MyFirstRobot  
Note that the name cannot be empty or contain spaces.

A próxima tela é a nomeação do pacote, no caso foi escolhida: **competição**. Depois clique em **OK**.



Package name for MeuPrimeiroRobo

Enter a short package name for your new robot and without spaces (lower-case letters are preferred).  
Your initials will work well here.  
Your robot will be put into this package to avoid name conflict with other robots.  
The package name is used to identify your robot(s) in the game, especially if you want to let your robot(s) participate in competitions like e.g. RoboRumble@Home.  
Hence, you should enter the same package name for all of your robots.  
Note that the package name cannot be empty or contain spaces.

## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

#### 1.4 Explicando o código gerado

Segue a explicação do código gerado automaticamente do nosso Robô.

```

1 package competicao;
2 import robocode.*;
3 //import java.awt.Color;
4
5 // API help : http://robocode.sourceforge.net/docs/robocode/robocode/Robot.html
6
7 /**
8  * MeuPrimeiroRobo - a robot by (your name here)
9  */
10 public class MeuPrimeiroRobo extends Robot
11 {
12     /**
13     * run: MeuPrimeiroRobo's default behavior
14     */
15     public void run() {
16         // Initialization of the robot should be put here
17
18         // After trying out your robot, try uncommenting the import at the top,
19         // and the next line:
20
21         // setColors(Color.red,Color.blue,Color.green); // body,gun,radar
22
23         // Robot main loop
24         while(true) {
25             // Replace the next 4 lines with your own code
26             ahead(100);
27             turnGunRight(360);
28             back(100);
29             turnGunRight(360);
30         }
31     }
32

```

Importa as classes do Robocode

Pacote criado. Em uma competição é recomendado que todas as equipes criem o robô indicando o mesmo nome do pacote.

Inclua aqui os dados da equipe

Classe criada aplicando herança, podendo utilizar todos os recursos da classe Robot

Método que define a inicialização e configuração do Robô

Looping da sequência de comandos para o robô executar.  
 ahead – a distância que o robô deverá percorrer para frente  
 turnGunRight – gira o canhão para a direita 360 graus  
 back – a distância que o robô deve percorrer para trás

## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

```

33  /**
34  * onScannedRobot: What to do when you see another robot
35  */
36  public void onScannedRobot(ScannedRobotEvent e) {
37      // Replace the next line with any behavior you want
38      fire(1);
39  }
40
41  /**
42  * onHitByBullet: What to do when you're hit by a bullet
43  */
44  public void onHitByBullet(HitByBulletEvent e) {
45      // Replace the next line with any behavior you want
46      back(10);
47  }
48
49  /**
50  * onHitWall: What to do when you hit a wall
51  */
52  public void onHitWall(HitWallEvent e) {
53      // Replace the next line with any behavior you want
54      back(20);
55  }
56 }
57

```

Método que verifica se algum robô entrou na mira, e realiza um tiro de nível 1. Quanto maior o nível do tiro, mais energia perde.

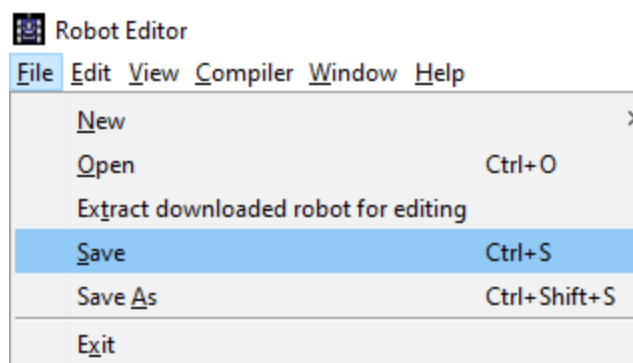
Método que verifica quando recebe um tiro

Método que verifica quando o robô atinge a parede

### 1.5 Salvando e Compilando o Robô

Agora que vimos o código gerado automaticamente de nosso robô, para compilar temos que realizar os seguintes passos:

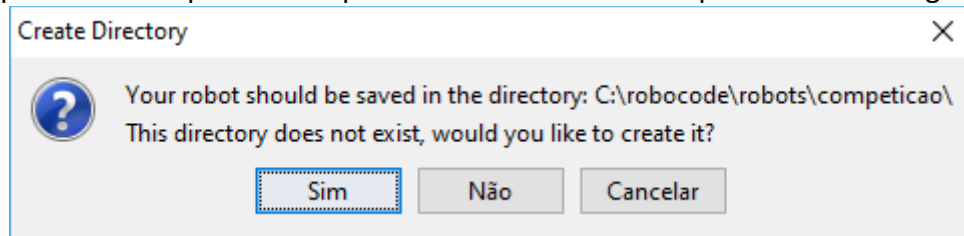
- 1) Clique na opção do menu **File** e depois em **Save**.



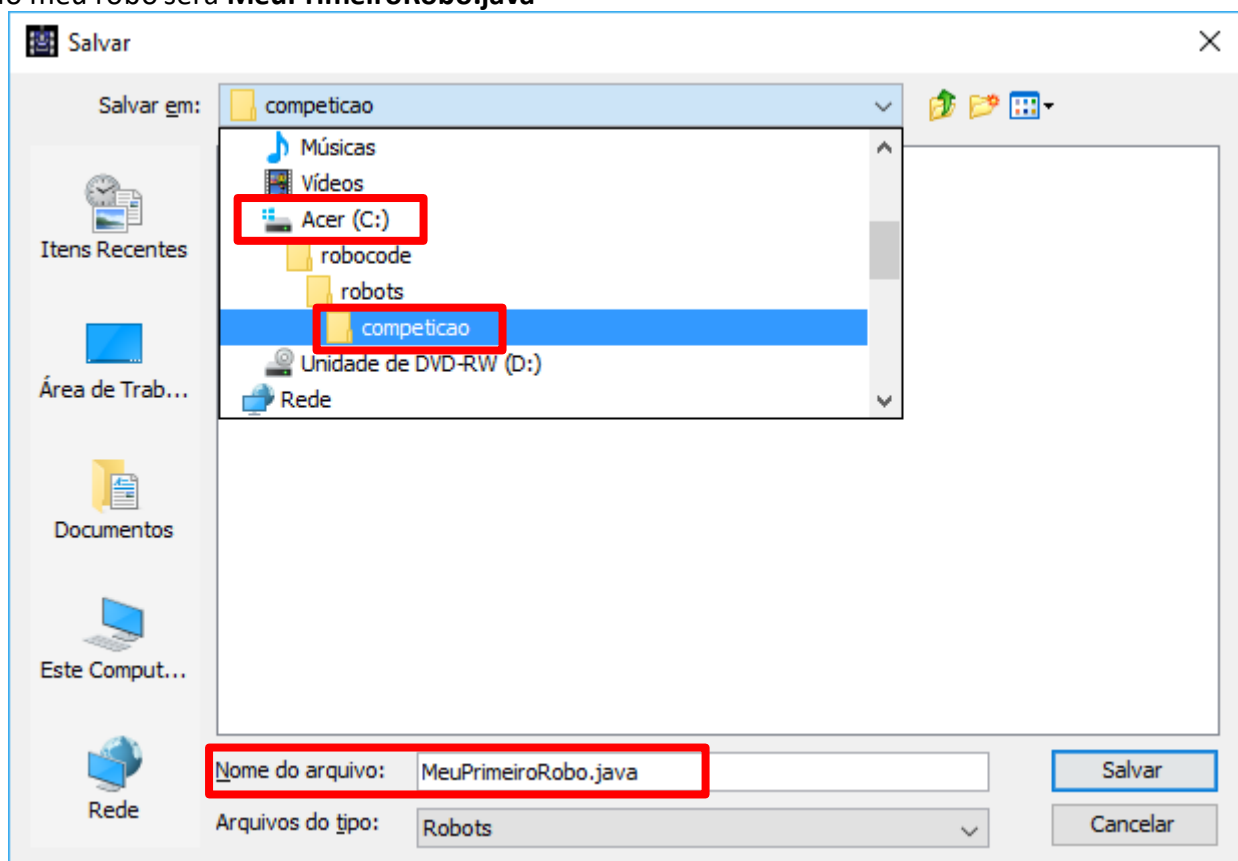
## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

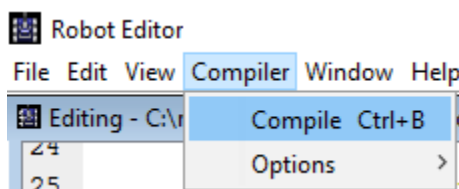
Agora, na janela que se abre clique em **SIM** para escolher um diretório para salvar o código:



Como você irá perceber na figura abaixo, meu robô ficará salvo na pasta **competicao**, que foi criada quando foi atribuído o nome do pacote do robô. Esta pasta está localizada em: **C:\robocode\robots\competicao** e o nome do meu robô será **MeuPrimeiroRobo.java**



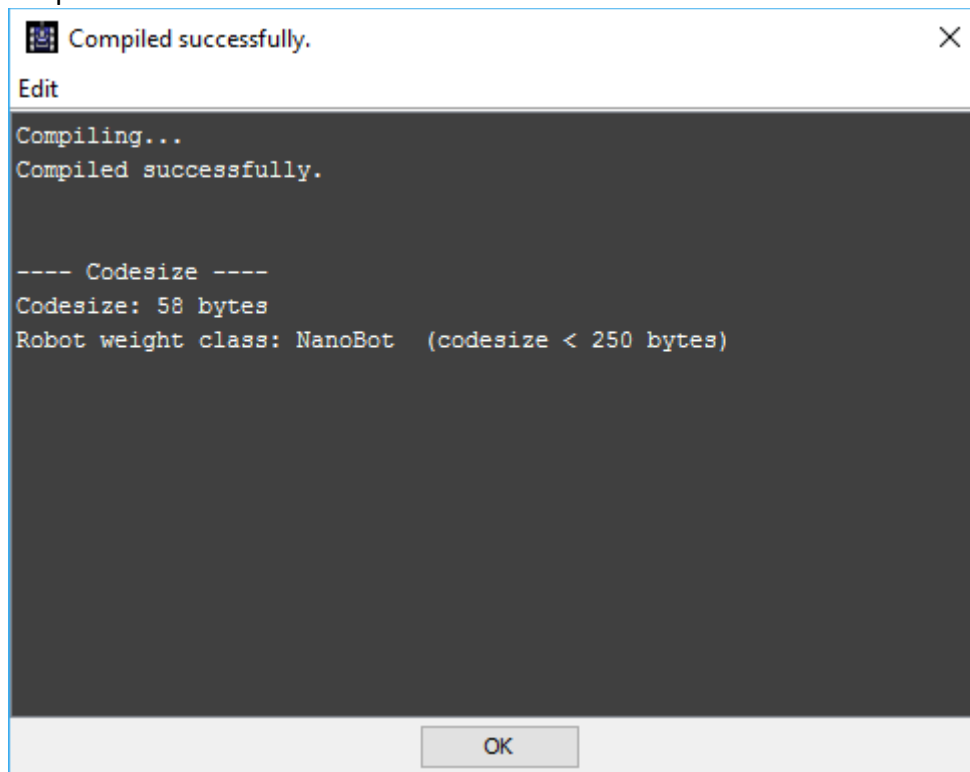
Agora que nosso código está salvo, precisamos compila-lo e para isso, temos que clicar no menu em **Compiler** e depois em **Compile**:



## Administração Central

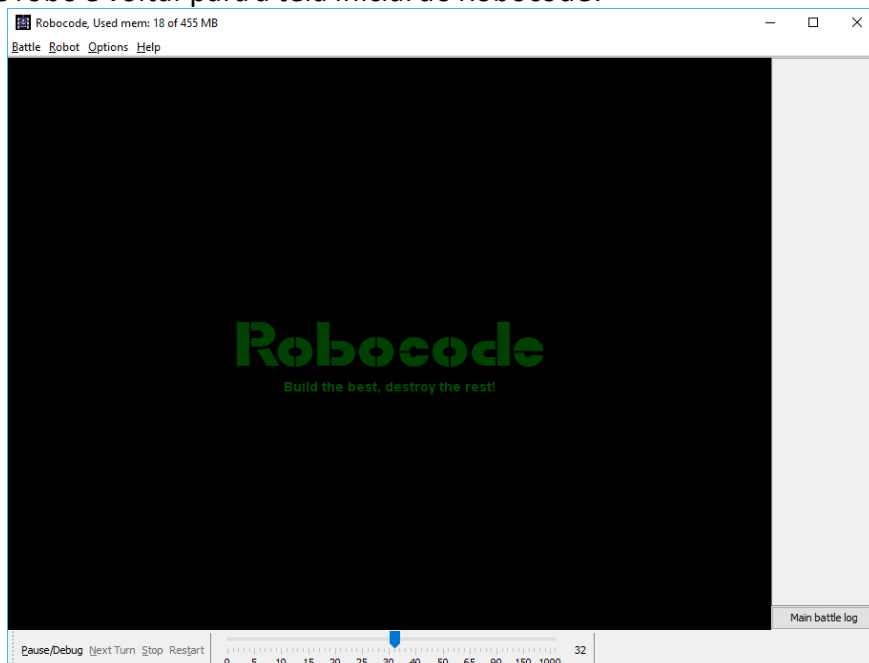
### Unidade do Ensino Médio e Técnico – Cetec Capacitações

O programa será compilado e será mostrado se foi compilado com sucesso ou não, como mostra a imagem a seguir. Clique em **OK** para sair da tela:



## 2 Colocando nosso Robô na arena

Pode fechar editor de robô e voltar para a tela inicial do Robocode.

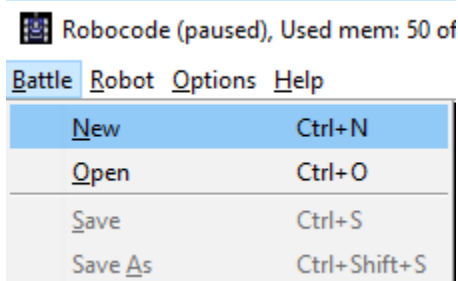




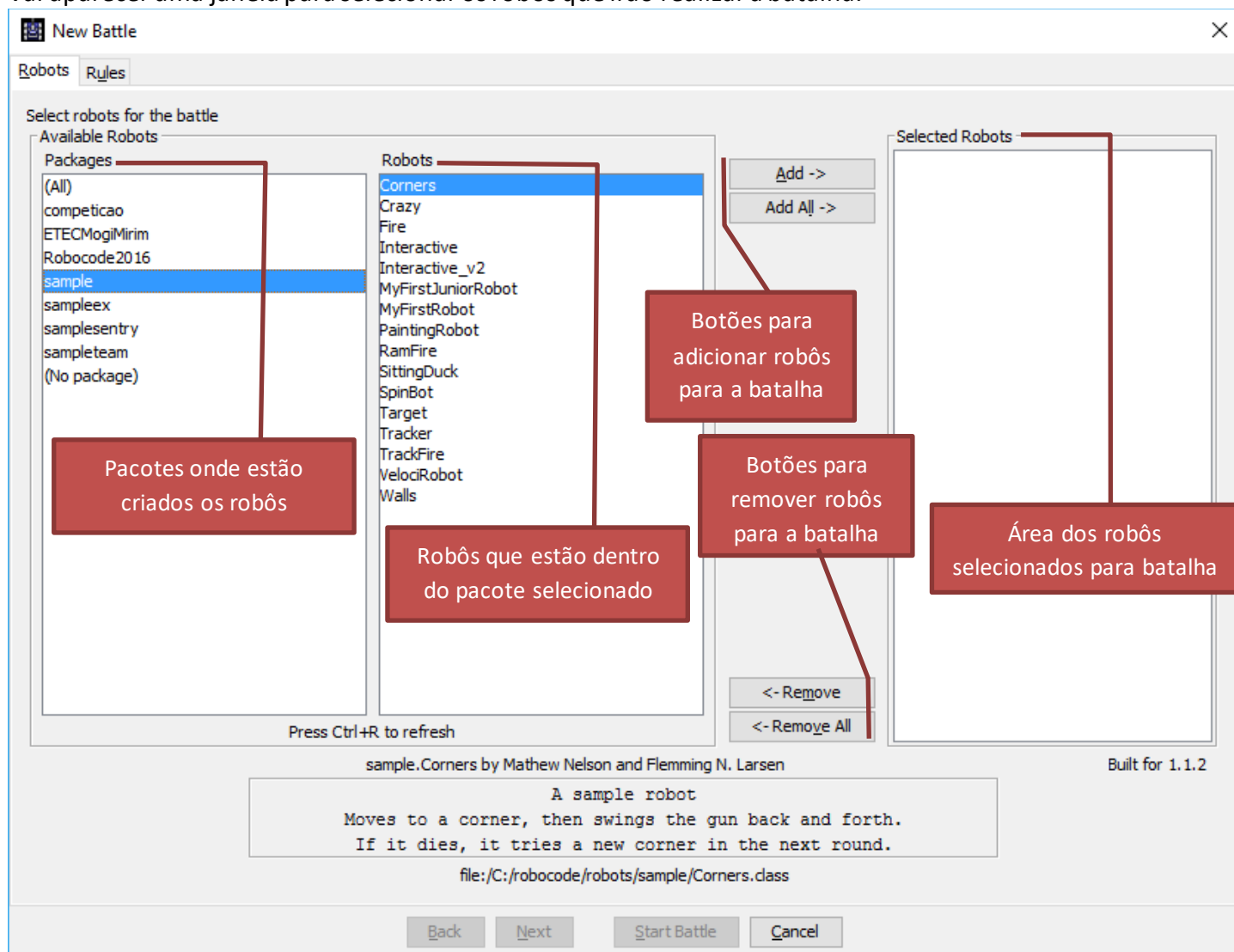
## Administração Central

Unidade do Ensino Médio e Técnico – Cetec Capacitações

Agora, vamos à nossa arena, clicando em **Battle**, depois em **New**.



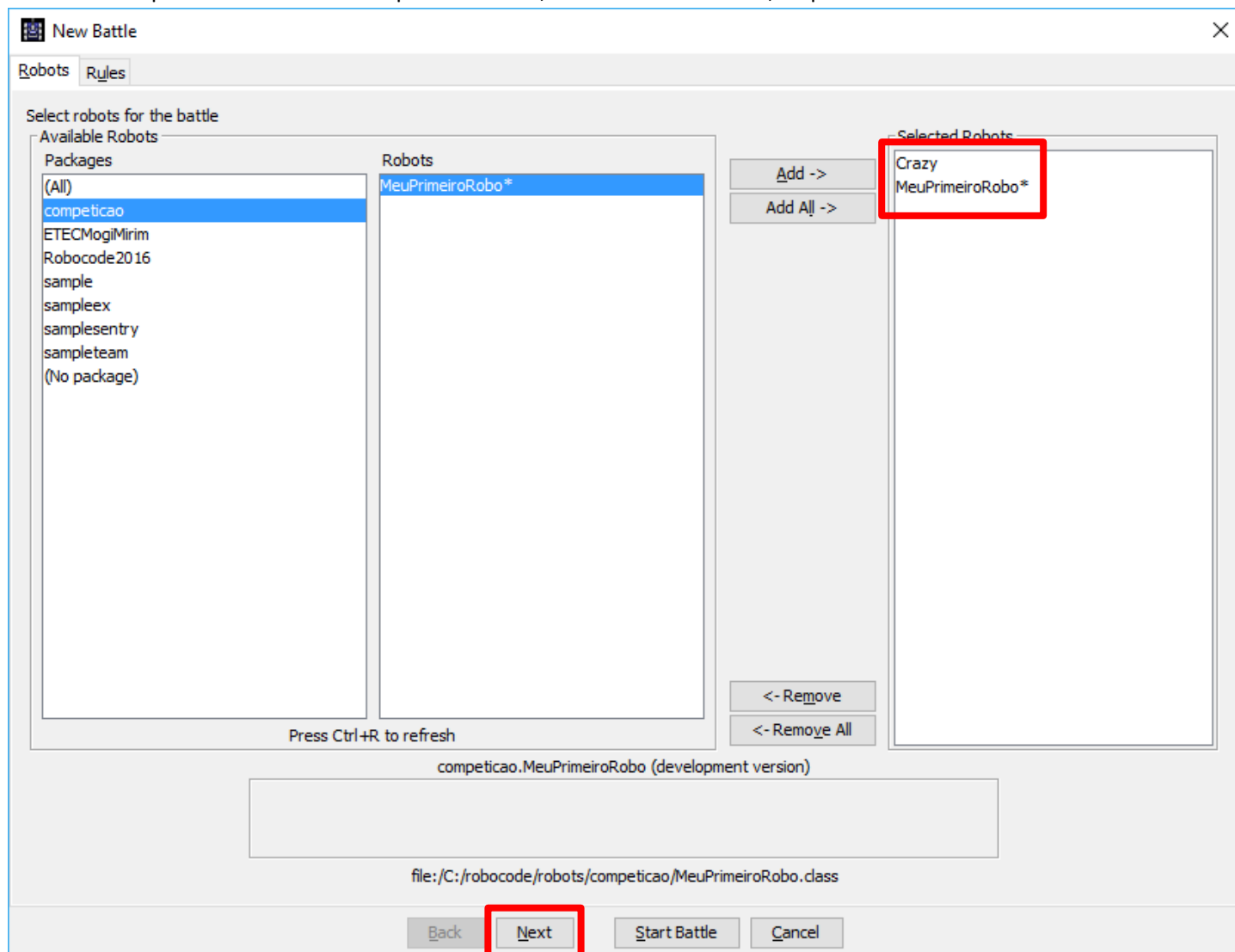
Vai aparecer uma janela para selecionar os robôs que irão realizar a batalha.



## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

Ficará assim após selecionar os robôs para a batalha, e estando tudo certo, clique em **NEXT**:



Após clicar em **Next**, seremos direcionados para a aba **Rules**, onde podemos definir algumas configurações da batalha.

**Number of Rounds:** Definimos o número de Rounds da batalha.

**Cooling Rate:** Definimos o intervalo de tempo do tiro (entre 0.1 e 0.7)

**Inactivity Time:** Definimos o tempo máximo que um robô pode ficar inativo no campo de batalha antes de executar alguma ação.

**Sentry Border Size:** Definimos uma cerca inserida no campo de batalha, que funciona somente com robôs que implementam a interface BorderSentry, que têm a função básica de agirem como defensores, limitando seus movimentos somente dentro da borda criada.

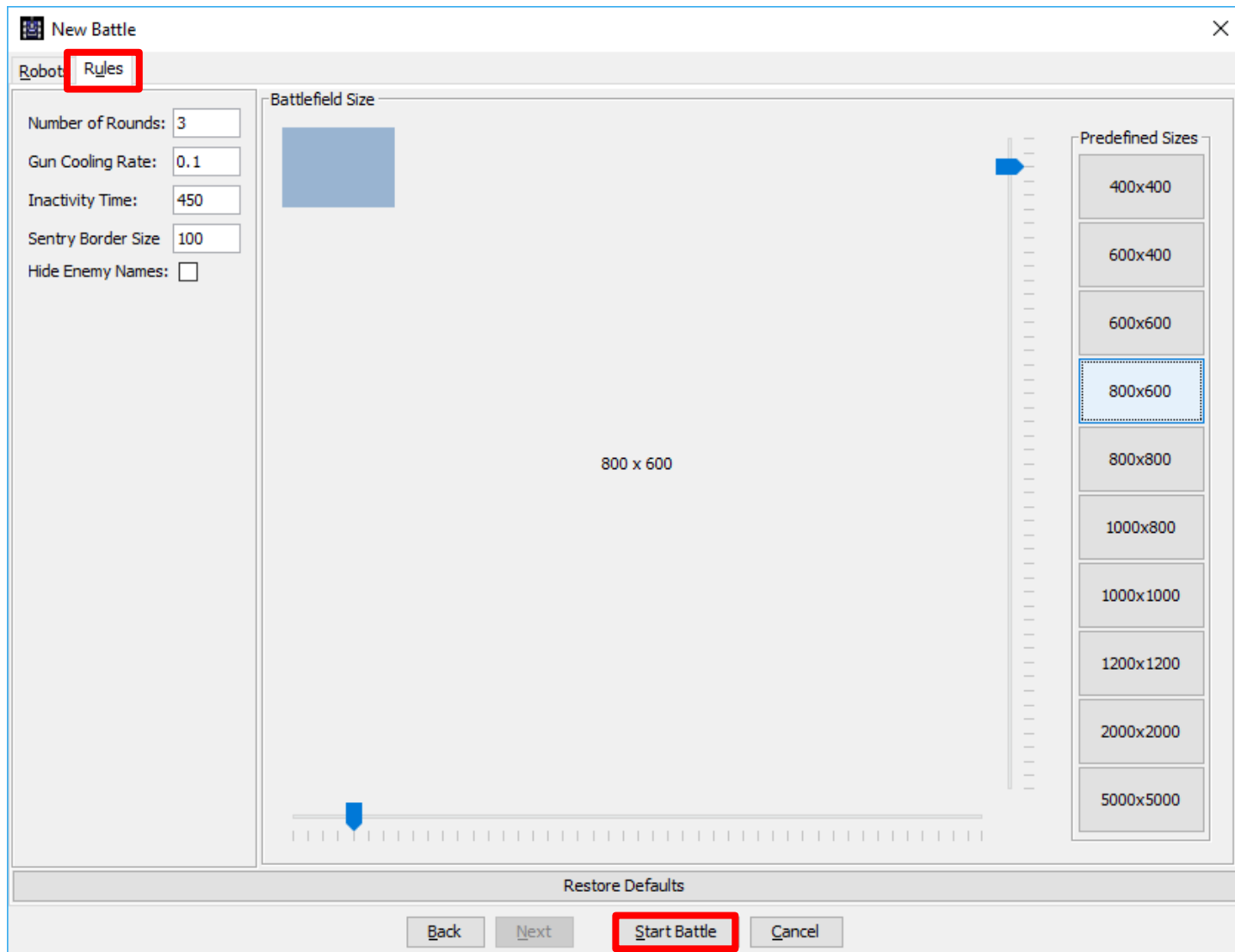
**Hide Enemy Names:** possui a função de ocultar os nomes dos oponentes que estão no campo de batalha.

**Preferred Sizes:** Definimos a dimensão do campo de batalha

[www.centropaulasouza.sp.gov.br](http://www.centropaulasouza.sp.gov.br)

## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações



**New Battle**

Robot: **Rules**

Number of Rounds: 3  
Gun Cooling Rate: 0.1  
Inactivity Time: 450  
Sentry Border Size: 100  
Hide Enemy Names:

Battlefield Size

800 x 600

Predefined Sizes

- 400x400
- 600x400
- 600x600
- 800x600**
- 800x800
- 1000x800
- 1000x1000
- 1200x1200
- 2000x2000
- 5000x5000

Restore Defaults

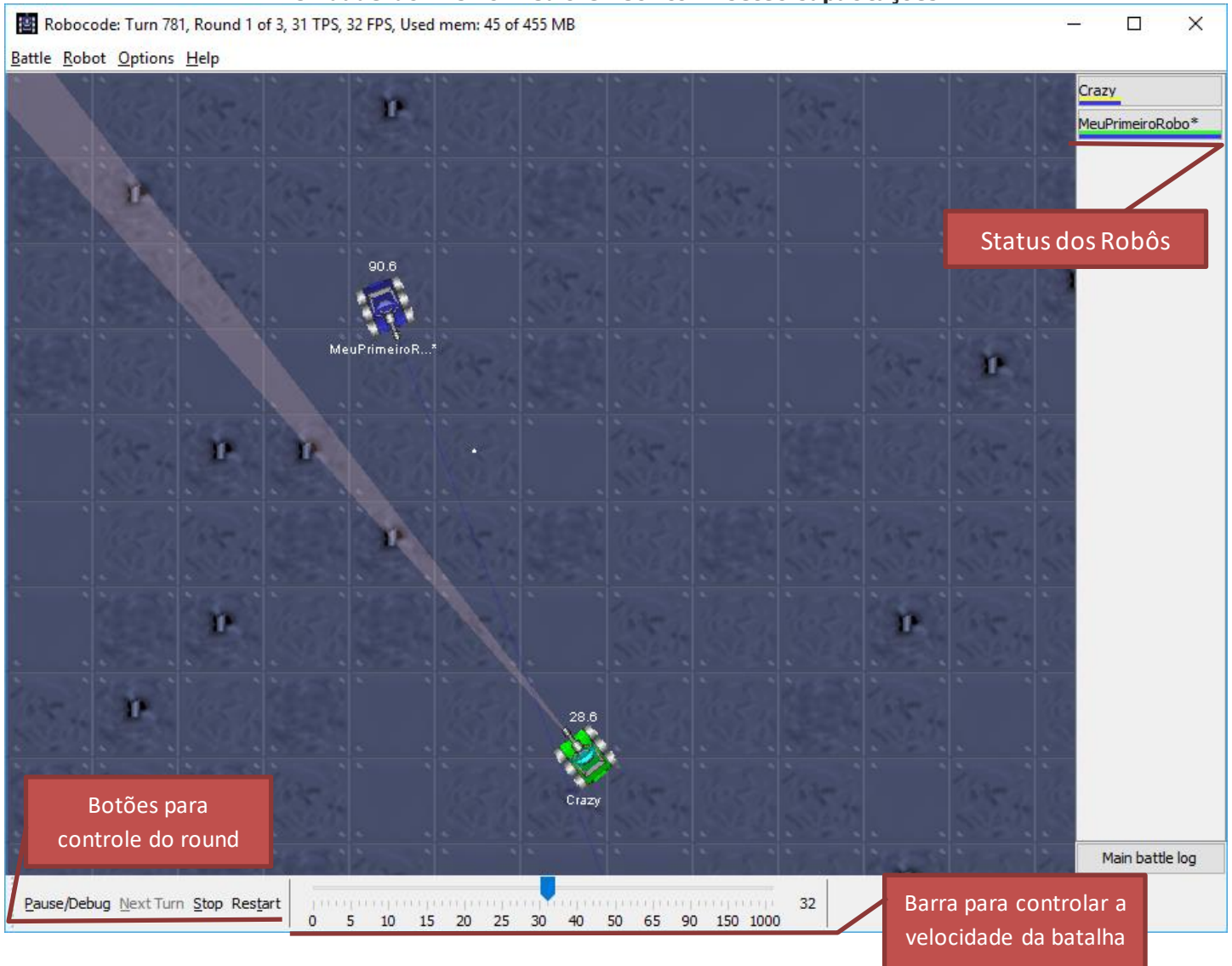
Back Next **Start Battle** Cancel

Obs: Como já devem ter percebido o Robocode já possui vários robôs prontos e isso é muito bom, pois podemos estudar os códigos dos robôs existentes.

Agora é só clicar em **Start Battle**, para iniciar a batalha.

## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações



Robocode: Turn 781, Round 1 of 3, 31 TPS, 32 FPS, Used mem: 45 of 455 MB

Battle Robot Options Help

90.6  
MeuPrimeiroR...\*

28.6  
Crazy

Botões para controle do round

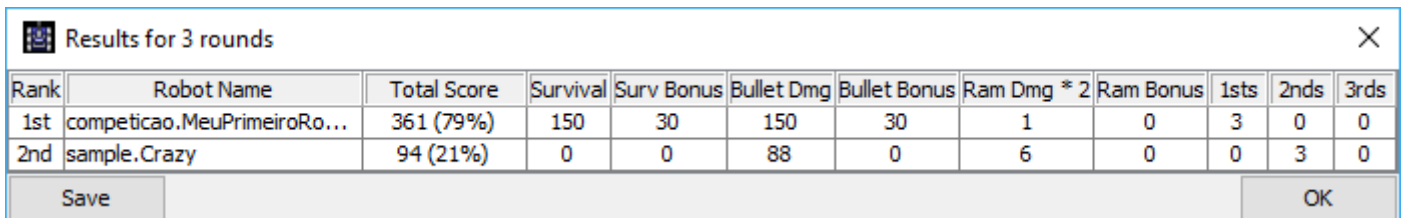
Status dos Robôs

Barra para controlar a velocidade da batalha

Pause/Debug Next Turn Stop Restart

Main battle log

Ao final de cada batalha será mostrado o resultado:



| Rank | Robot Name                  | Total Score | Survival | Surv Bonus | Bullet Dmg | Bullet Bonus | Ram Dmg * 2 | Ram Bonus | 1sts | 2nds | 3rds |
|------|-----------------------------|-------------|----------|------------|------------|--------------|-------------|-----------|------|------|------|
| 1st  | competicao.MeuPrimeiroRo... | 361 (79%)   | 150      | 30         | 150        | 30           | 1           | 0         | 3    | 0    | 0    |
| 2nd  | sample.Crazy                | 94 (21%)    | 0        | 0          | 88         | 0            | 6           | 0         | 0    | 3    | 0    |

Save OK

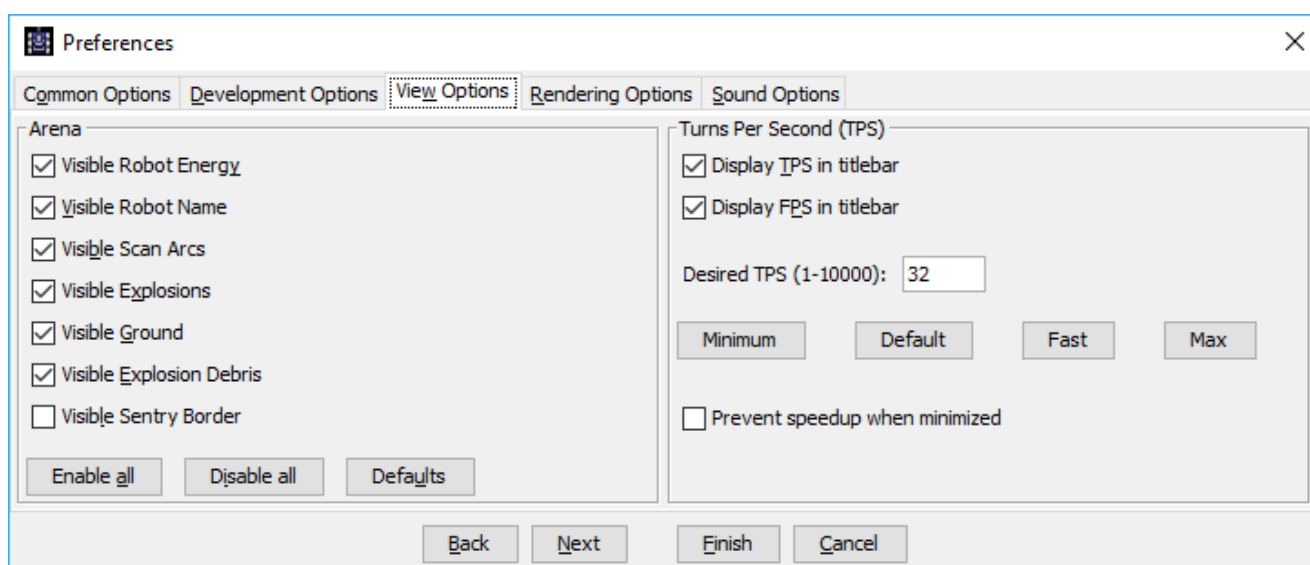
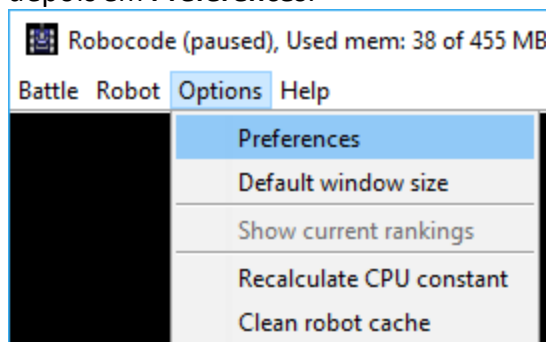
### 3 Opções e Visualizações

É muito importante neste período de iniciação na área do Robocode, habilitar o SCAN, para que vocês possam enxergar qual é a área de busca do robô.

## Administração Central

Unidade do Ensino Médio e Técnico – Cetec Capacitações

Para isso, clique em **Options** e depois em **Preferences**.



## 4 Modificando o Código

Só para fazer um teste, vamos usar o robô que criamos e retirar parte do código para analisarmos as alterações:

Para modificar seu primeiro robô, no menu clique em **Robot**, depois em **Source Editor**. Na tela do editor, no menu clique em **File**, depois em **Open** e escolher o robô.

No código aberto, procure o comando WHILE e zere todos os parâmetros, como o exemplo a seguir:

```
while (true) {
    // Replace the next 4 lines with any behavior you would like
    ahead(0);
    turnGunRight(0);
    back(0);
    turnGunRight(0);
}
```

---

## Administração Central

### Unidade do Ensino Médio e Técnico – Cetec Capacitações

Agora, devemos salvar clicando em **File** e depois em **Save**. E como já fizemos anteriormente, devemos compilar o código clicando em **Compiler** e depois em **Compile**.

Coloque agora novamente os dois robôs na arena. Com estas modificações, podemos notar que nosso robô não procura mais pelo robô inimigo e que ele só se movimenta se for atingido por algum tiro. Agora, é a hora de iniciar os testes e modificar um parâmetro por vez para ir cada vez mais aprendendo sobre o robô. Mãos a obra!

## 5 Referências

[1] Autor Desconhecido. **GSIGMA – Universidade Federal de São Carlos**. Disponível em:

<http://www.gsigma.ufsc.br/~popov/aulas/robocode/metodos.html> Acesso em 12 de março de 2019.

[2] Autor Desconhecido. **Site Oficial ROBOCODE**. Disponível em: <http://robocode.sourceforge.net/> Acesso em 11 de março de 2019.