

Construindo um Robô

Equipe de Robótica

Profa. Jacqueline

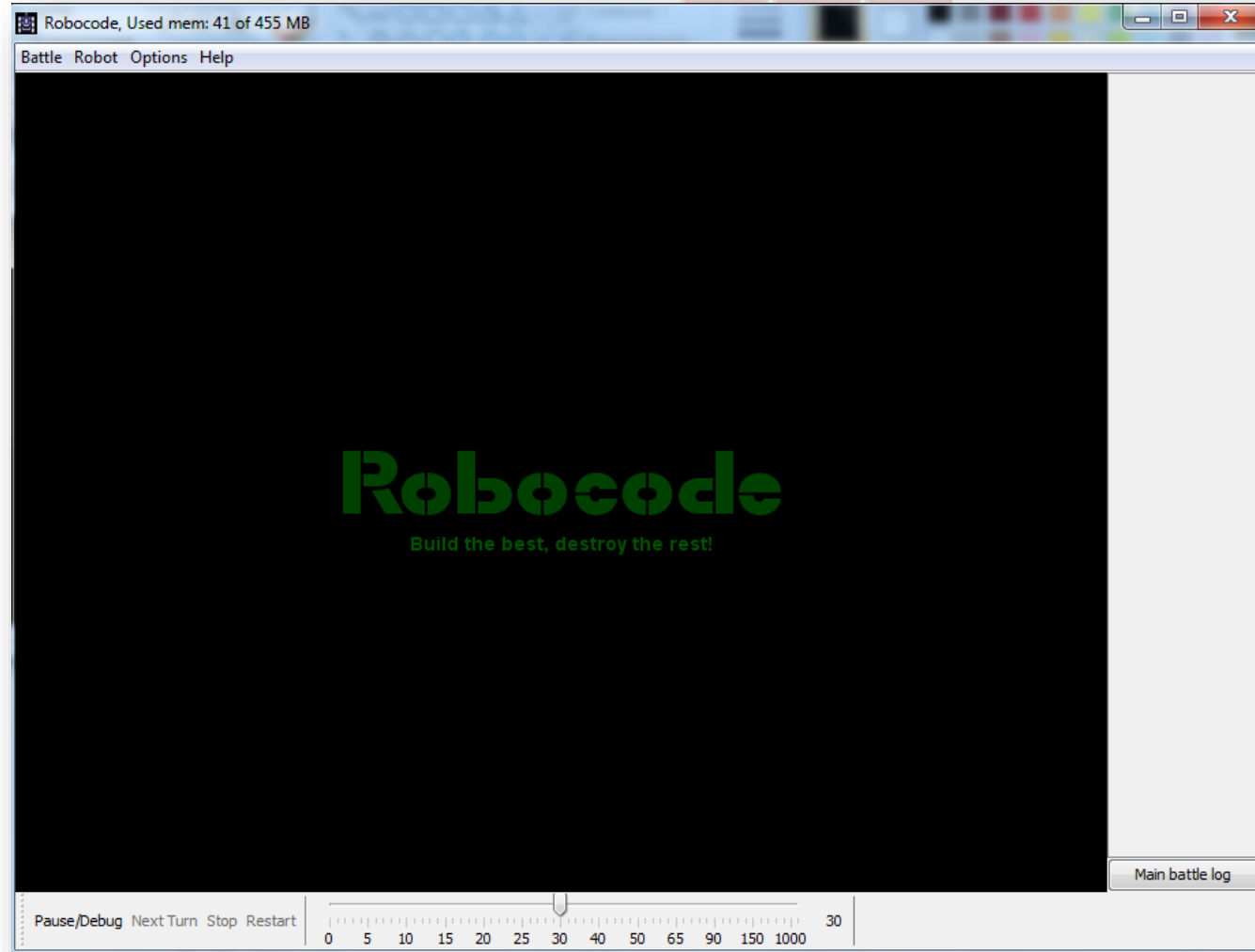
Prof. Sidnei

Prof. Fabrício

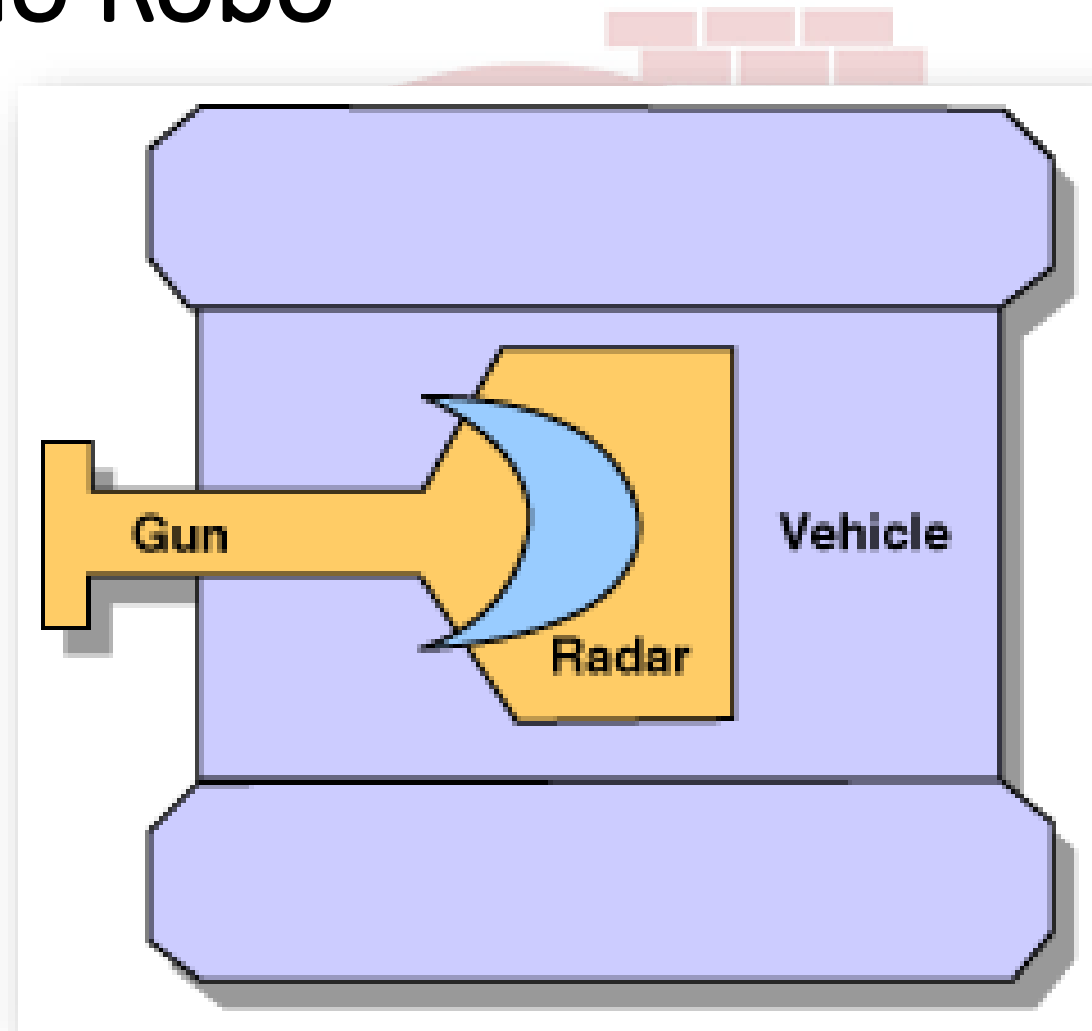
CONSTRUIR JUNTOS



Ambiente de Batalha



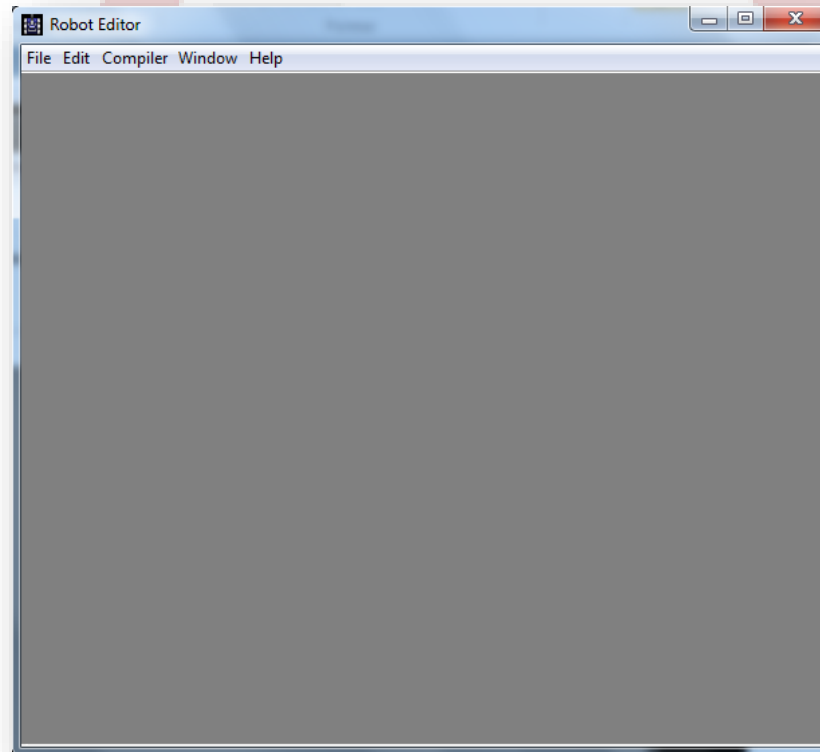
Anatomia do Robô





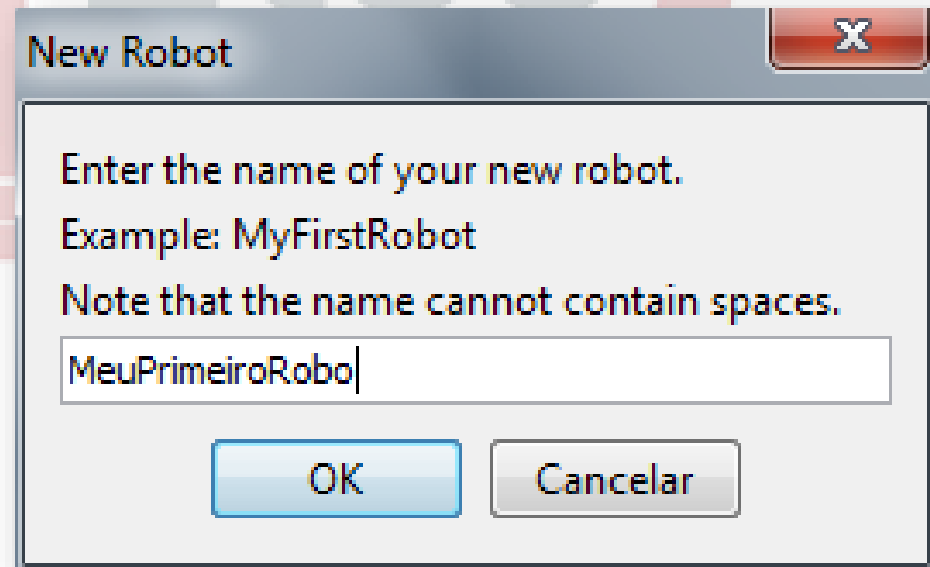
Criando o primeiro robô

- Para criar seu primeiro robô, clique no Menu **Robot** e depois em **Source Editor** **Ctrl+E**, em seguida abra a tela:



Criando o primeiro robô

- No Editor que abriu clique em **File**, e depois em **New**
depois em **Robot** **Ctrl+N**
- Como este é o nosso primeiro robô, estou nomeando de **MeuPrimeiroRobo** e clique em **OK** (Obs:- Sempre inicie o nome em **MAIÚSCULO**).





- Na tela seguinte é preciso nomear o pacote, neste caso caso digitamos **etec_Montemor**.

A screenshot of a dialog box titled "Package name for MeuPrimeiroRobo". The dialog box contains instructions for naming a package: "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." Below the text is a text input field containing the text "etec_montemor". At the bottom of the dialog box are two buttons: "OK" and "Cancelar".

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.

Código da Programação

- A tela a seguir mostra o código básico gerado automaticamente.

```
1 package etec_montemor;
2 import robocode.*;
3 //import java.awt.Color;
4
5 // API help : http://robocode.sourceforge.net/docs/robocode
6
7 /**
8  * MeuPrimeiroRobo - a robot by (your name here)
9  */
10 public class MeuPrimeiroRobo extends Robot
11 {
```

Nome do pacote que nomeamos inicialmente

Importar as classe **robocode**

Coloque aqui o nome da sua equipe

Classe criada MeuPrimeiroRobo com extend para usar métodos e eventos

Código da Programação

```
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, g
22
23      // Robot main loop
24      while(true) {
25          // Replace the next 4 lines with any behavior you wou
26          ahead(100);
27          turnGunRight(360);
28          back(100);
29          turnGunRight(360);
30      }
31  }
```

Método Run determina o posicionamento do robô, inicialização e configuração.

Note que também tem um laço de repetição para repetir os métodos:

```
ahead(100);
turnGunRight(360);
Back(100);
turnGunRight(360);
```


Código da Programação

```
32
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
```

Método que verifica se algum Robô entrou na mira, e dá um tiro de força 1. Este gasta pouca energia, mas tem tiros mais fortes, porém, quanto mais forte, mais energia ele gasta.

Método que verifica quando recebo um tiro.



Código da Programação

```
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 }
```

Método que verifica quando o robô atinge a parede.

CONSTRUIR JUNTOS



Salvando o Robô

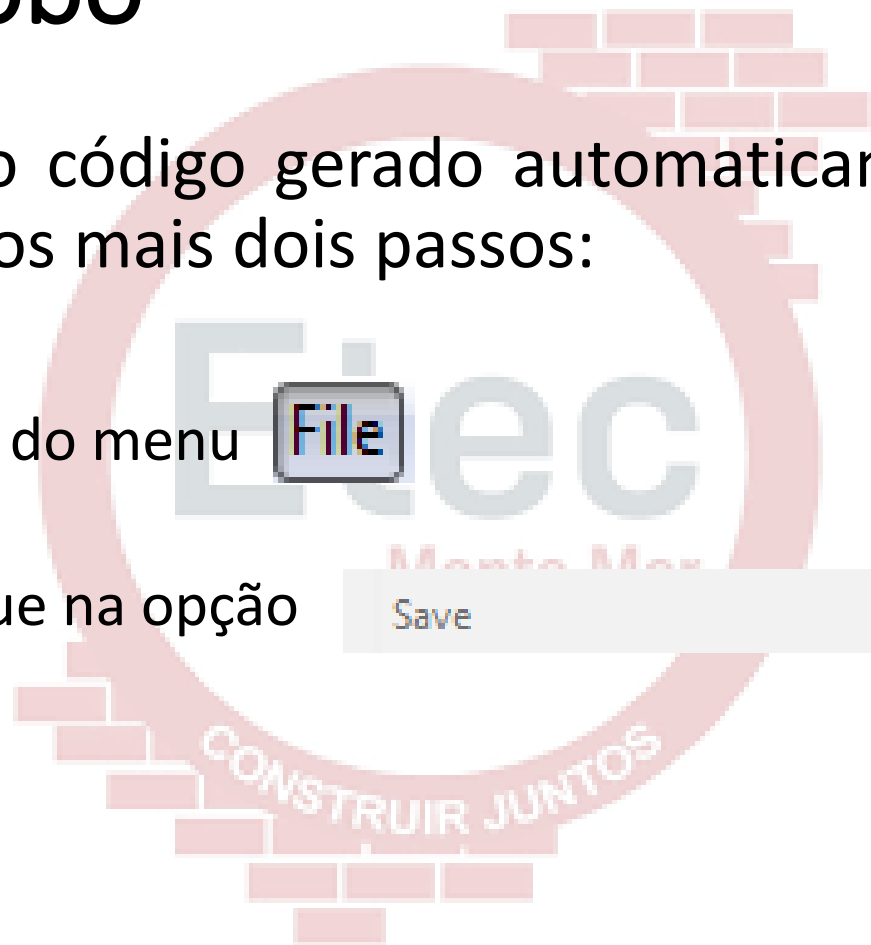
- Agora que vimos o código gerado automaticamente de nosso robô, para executar, temos mais dois passos:

1. Clique na opção do menu 

2. Em seguida clique na opção

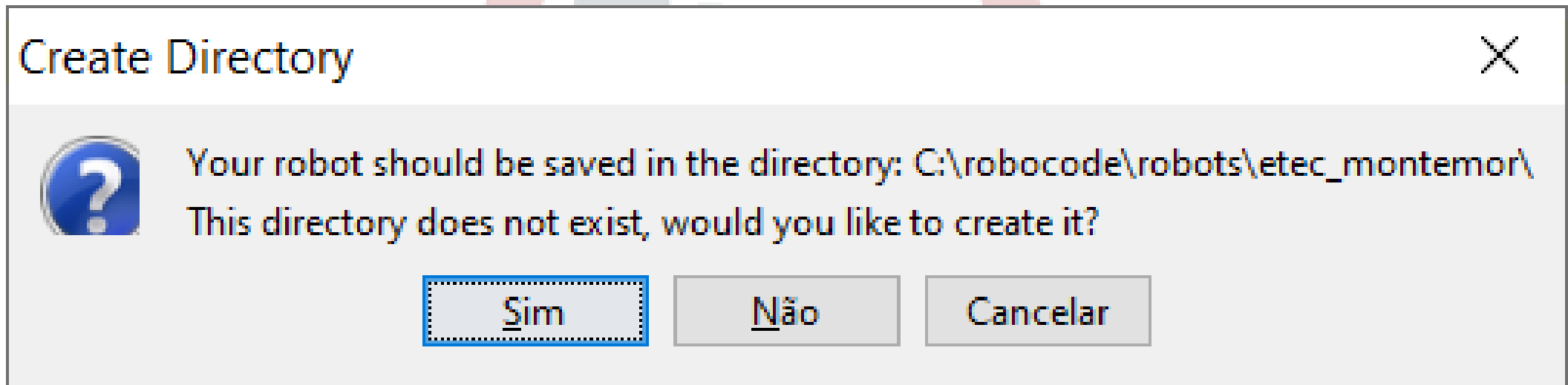
Save

Ctrl+S



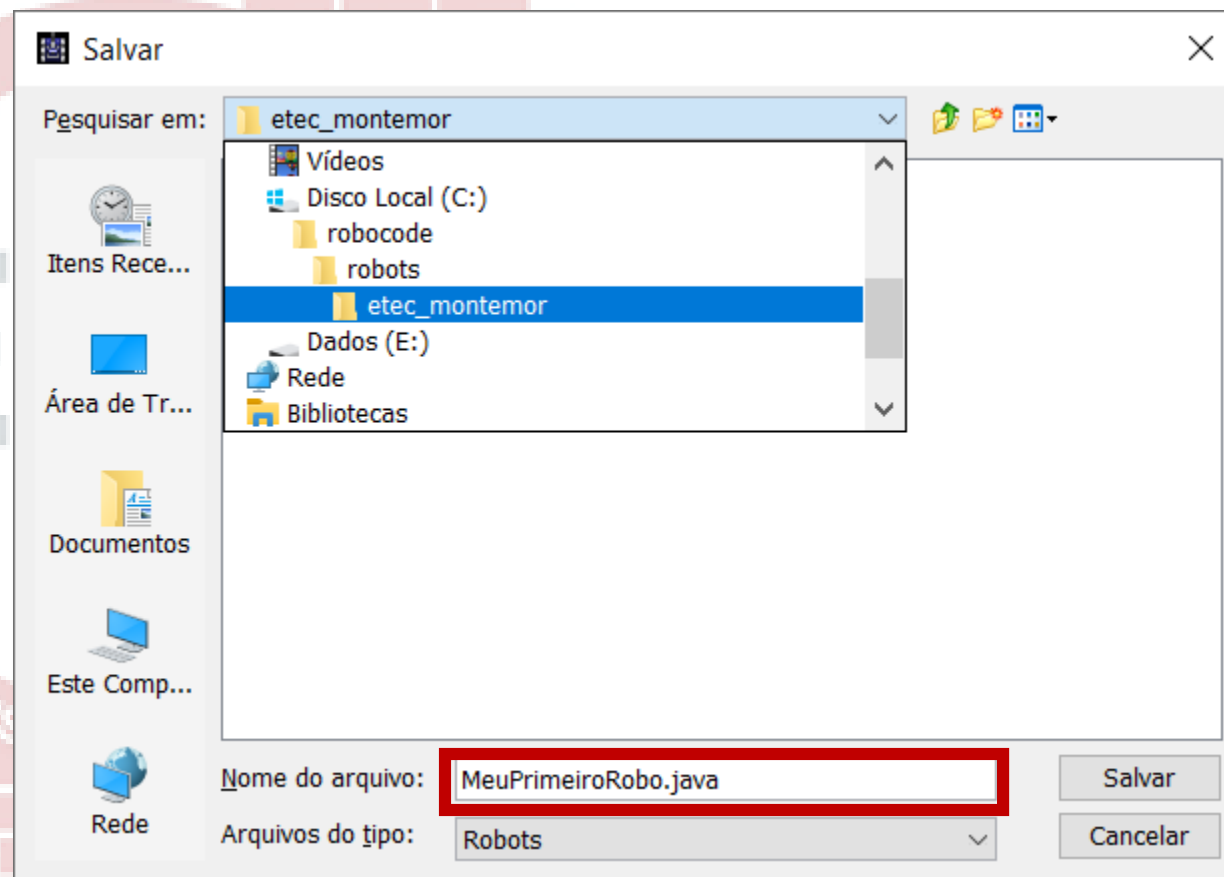
Salvando o Robô

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



Salvando o Robô

- Com mostra a figura, meu robô ficará salvo na pasta **etec_montemor**, que foi criada quando eu dei o nome do pacote do meu robô. Esta pasta está localizada em:
C:\robocode\robots\etec_montemor e o nome do meu robô será **MeuPrimeiroRobo.java**





Compilando o Robô

- Agora que nosso código está salvo, precisamos compila-lo e para isso temos mais dois passos:

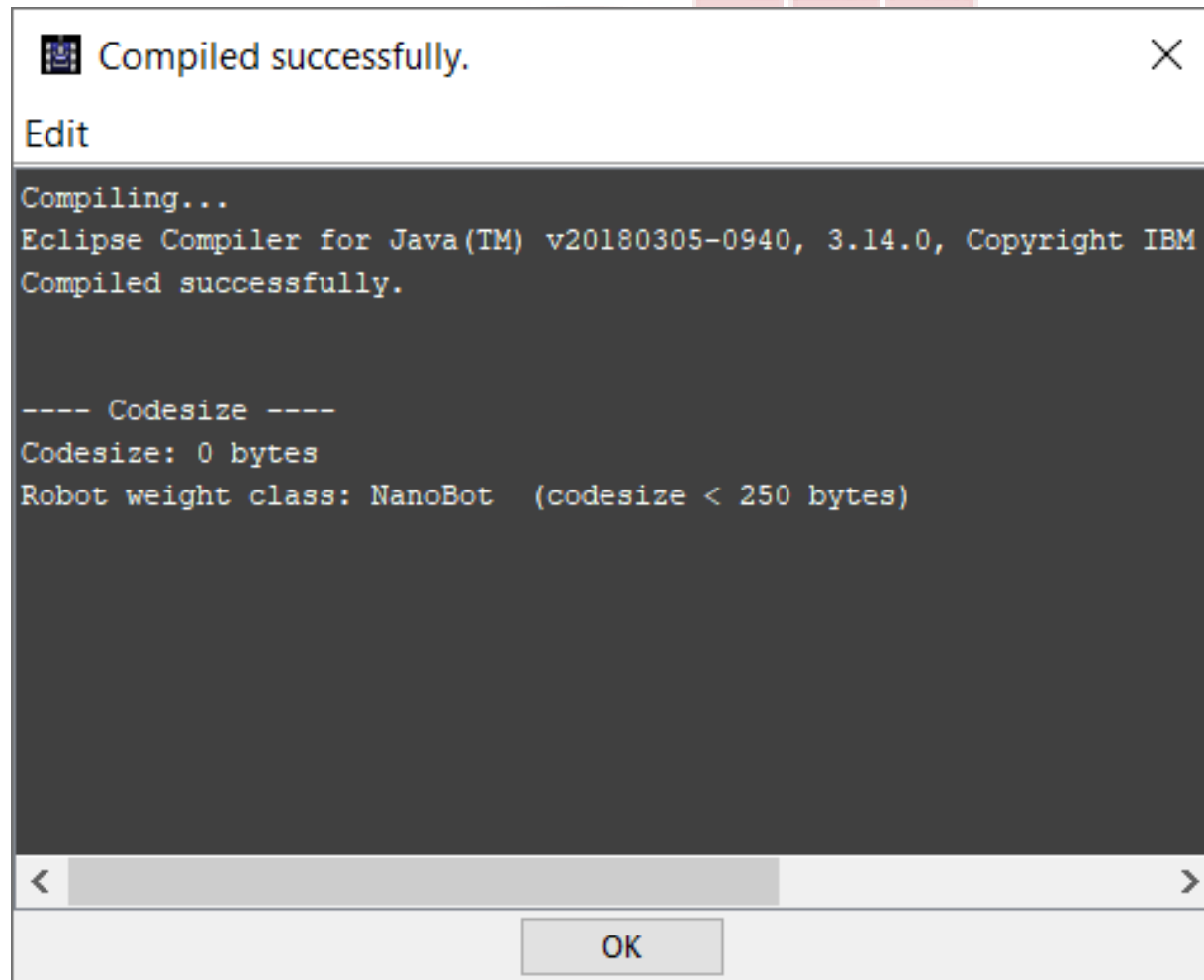
1. Clique na opção de menu 

2. Em seguida clique na opção 

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:



Compilando o Robô

A screenshot of a Java compilation dialog box. The title bar reads "Compiled successfully." with a close button on the right. Below the title bar is an "Edit" button. The main area is a dark text field containing the following text:

```
Compiling...  
Eclipse Compiler for Java(TM) v20180305-0940, 3.14.0, Copyright IBM  
Compiled successfully.  
  
---- Codesize ----  
Codesize: 0 bytes  
Robot weight class: NanoBot (codesize < 250 bytes)
```

At the bottom of the dialog box is an "OK" button.



Colocando o Robô na arena

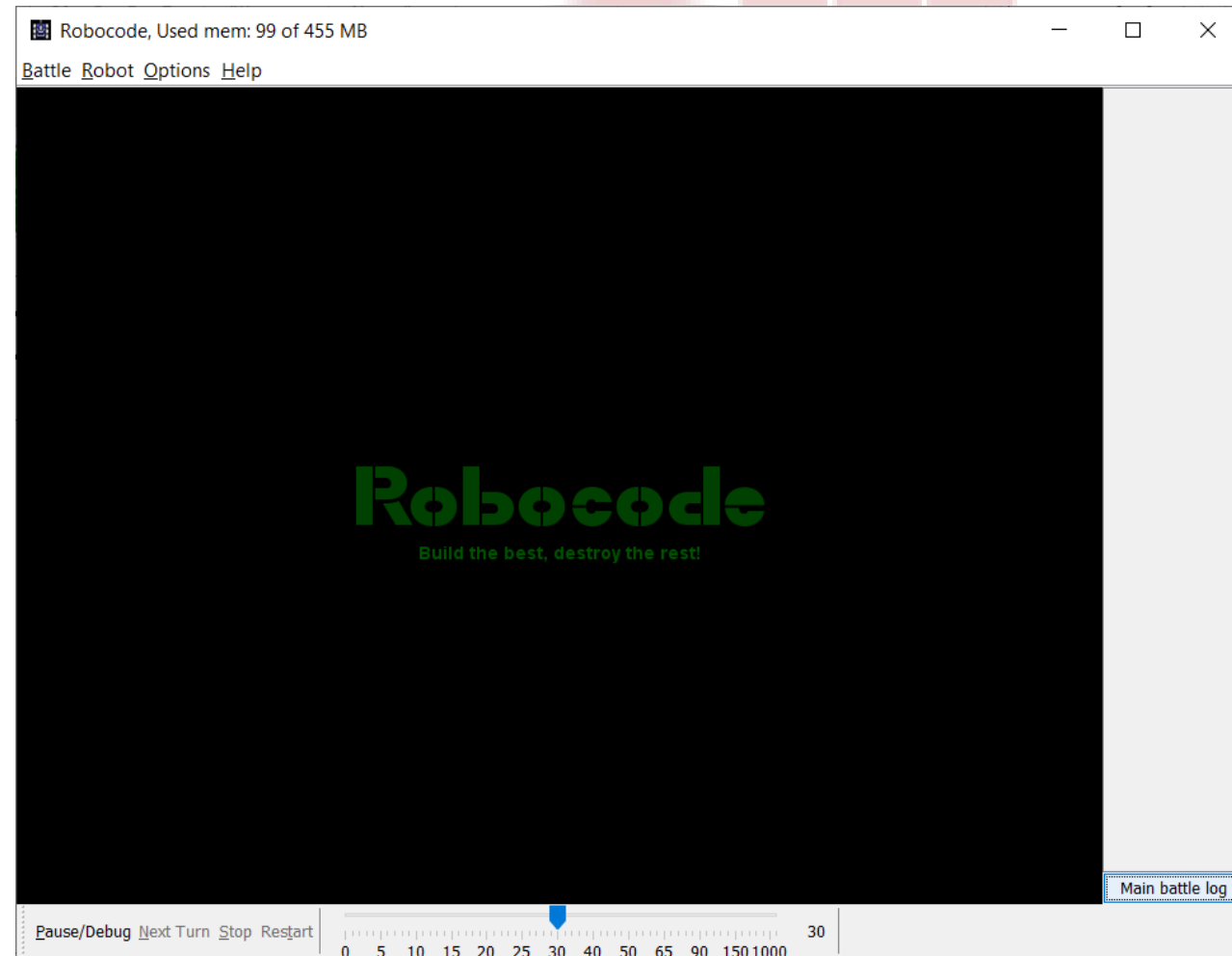
- O editor do robô pode ser fechado, caso você deseje.

```
Robot Editor
File Edit View Compiler Window Help
Editing - C:\robocode\robots\etec_montemor\MeuPrimeiroRobo.java
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 any behavior you would like
26          ahead(100);
27          turnGunRight(360);
28          back(100);
29          turnGunRight(360);
30      }
31  }
32
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 would like
38      fire(1);
39  }
```



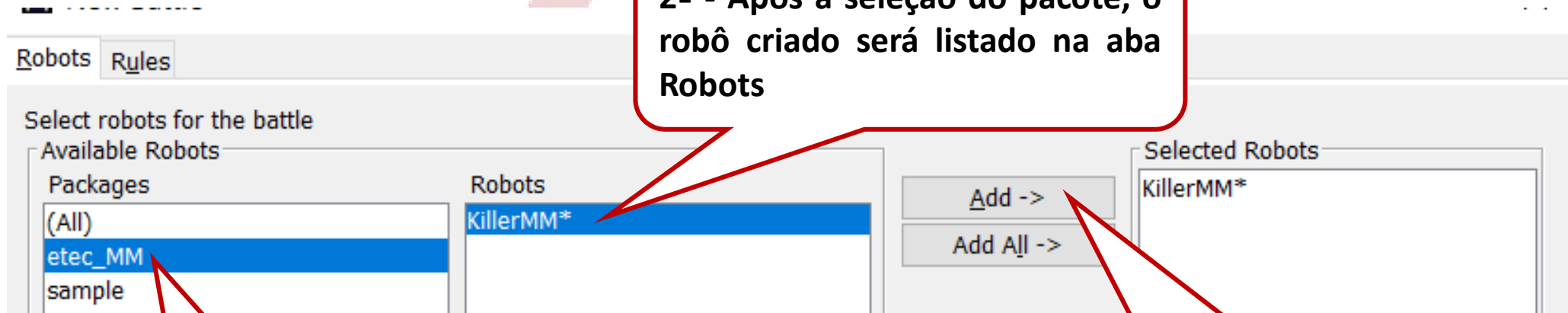

Colocando o Robô na arena

- A Arena



Colocando o Robô na arena

- Na arena, selecione **Battle** e em seguida **New** **Ctrl+N** , na janela que se abre, como



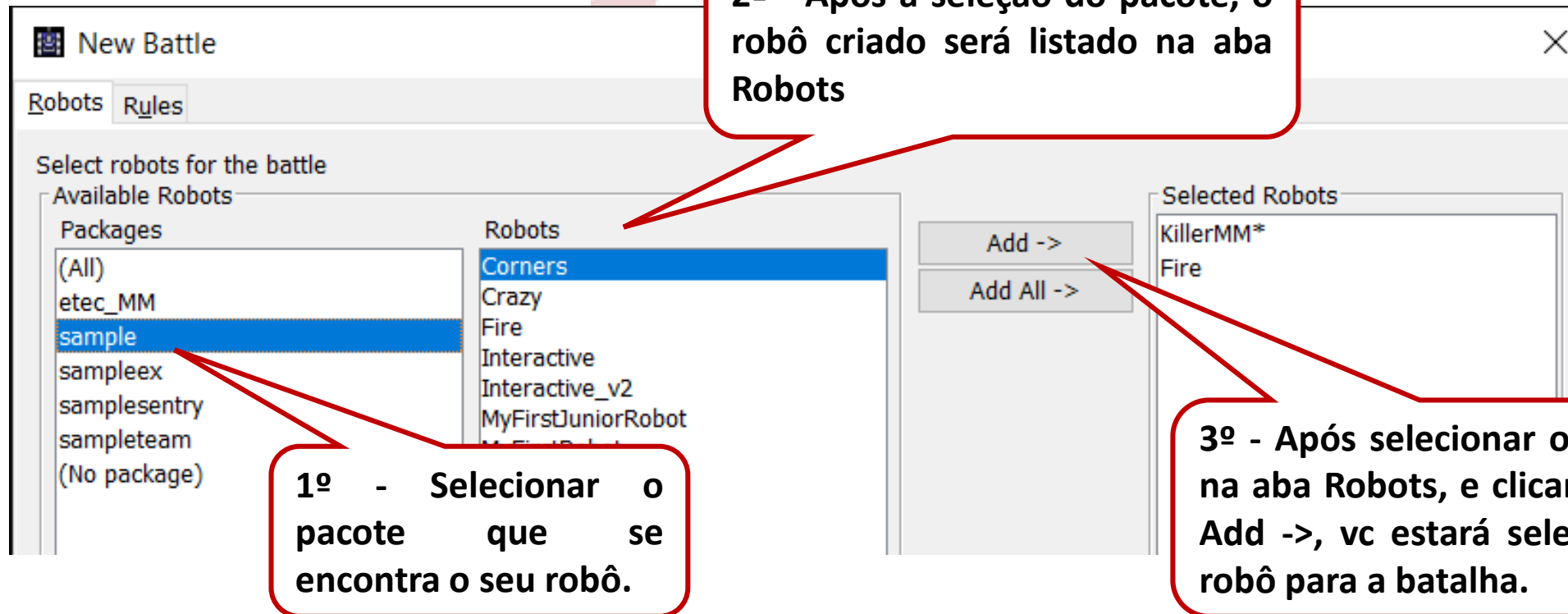
2º - Após a seleção do pacote, o robô criado será listado na aba Robots

1º - Selecionar o pacote que se encontra o seu robô.

3º - Após selecionar o seu robô, na aba Robots, e clicar no botão Add ->, vc estará selecionando o robô para a batalha.

Colocando o Robô na arena

- Para selecionar o(s) seu(s) oponente(s) basta repetir o mesmo processo anterior e escolher o(s) robô(s) que deseja.



The screenshot shows the 'New Battle' dialog box with two tabs: 'Robots' and 'Rules'. The 'Robots' tab is active, showing a list of available robots and a list of selected robots. The 'Available Robots' list is divided into 'Packages' and 'Robots'. The 'sample' package is selected, and the 'Corners' robot is selected in the 'Robots' list. The 'Selected Robots' list contains 'KillerMM*' and 'Fire'. The 'Add ->' button is highlighted, and the 'Add All ->' button is also visible.

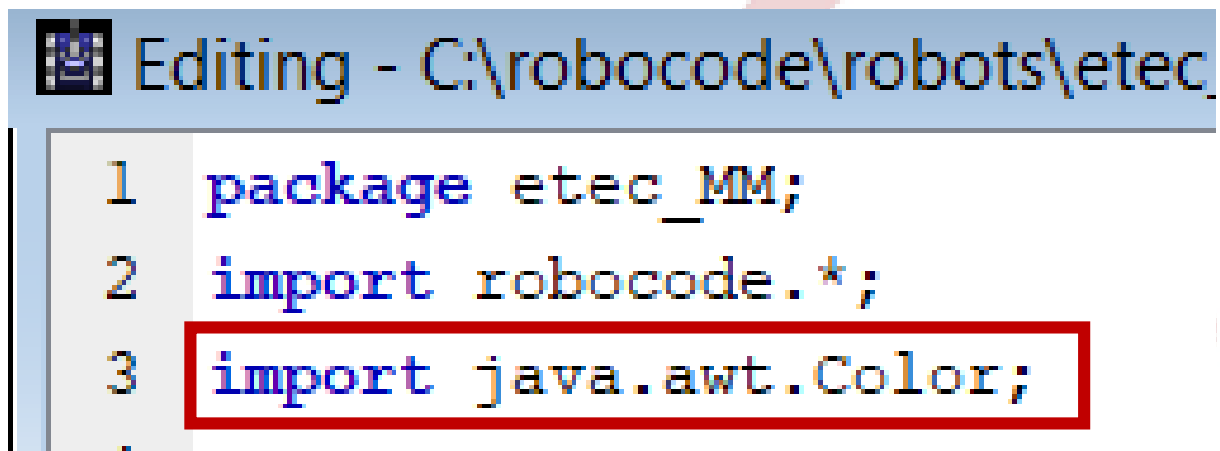
1º - Selecionar o pacote que se encontra o seu robô.

2º - Após a seleção do pacote, o robô criado será listado na aba Robots

3º - Após selecionar o seu robô, na aba Robots, e clicar no botão Add ->, vc estará selecionado o robô para a batalha.

Mudando a cor do robô

- Primeiro devemos a classe de cor.

A screenshot of a code editor window titled "Editing - C:\robocode\robots\etec...". The code is as follows:

```
1 package etec_MM;  
2 import robocode.*;  
3 import java.awt.Color;  
.
```

The third line, `import java.awt.Color;`, is highlighted with a red rectangular border. In the background, there is a large, faint watermark of a gear with the text "CONSTRUIR JUNTOS" and a large letter "C" with "Mor" below it.

Mudando a cor do Robô

- Em seguida vamos programar qual a parte do robô que deseja alterar a cor, através da função **setColors()**;

```
// After trying out your robot, try uncommenting the import at the  
// and the next line:
```

```
setColors(Color.red,Color.blue,Color.green); // body,gun,radar
```

body (Corpo
do Robô)

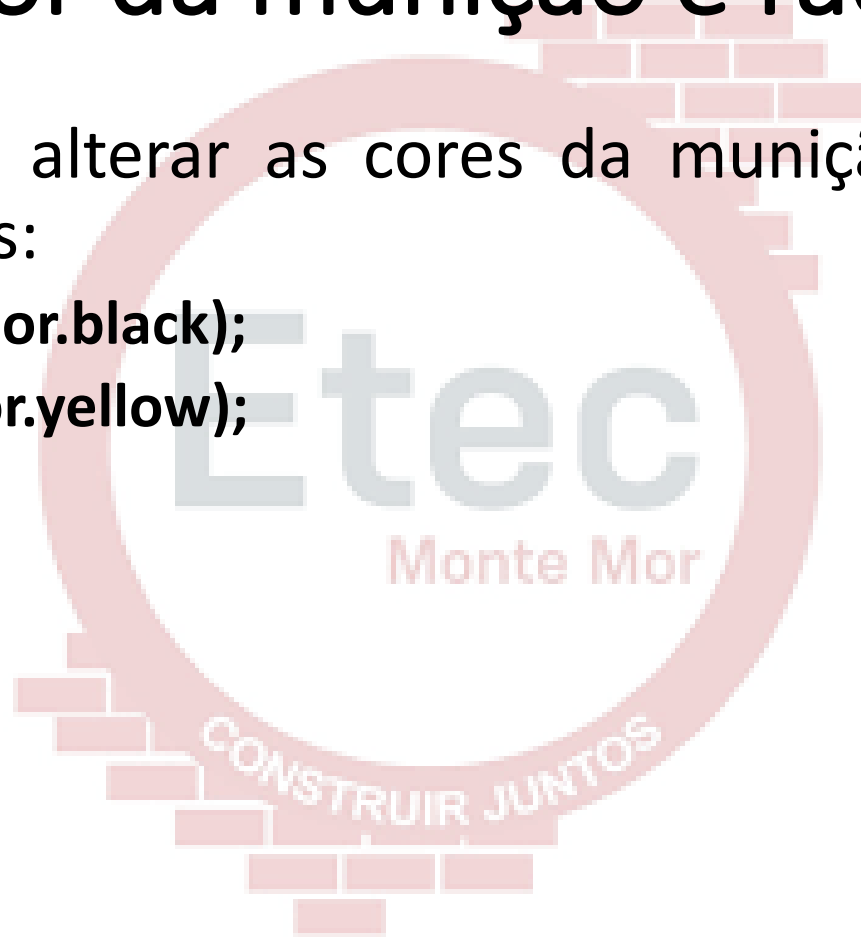
gun (Canhão
do Robô)

radar (Radar
do Robô)



Mudando a cor da munição e radar (Scan)

- Podemos também alterar as cores da munição e do radar (Scan), através das funções:
 - `setBulletColor(Color.black);`
 - `setScanColor(Color.yellow);`



Programando Eventos do Robô

- Eventos são as ações que o robô pode ter, tais como:
 - **Evento run()**
 - O Evento run(), é o evento básico no Robocode, por isso que quando criamos um robô, ele vem criado no código fonte:

```
public void run() {  
  
    while(true) {  
        ahead(100);  
        turnGunRight(360);  
        back(100);  
        turnGunRight(360);  
    }  
}
```
- Este evento, é o básico para fazer com que o tanque ande a frente, gire, atire e retorne



Programando Eventos do Robô

- **Evento onScannedRobot()**

- O Evento onScannedRobot(), é o evento que verifica se há algum robô no scanner, se tiver ele atira:

```
public void onScannedRobot (ScannedRobotEvent e) {  
    fire(1);  
}
```


Programando Eventos do Robô

- **Evento onHitByBullet()**

- O Evento onHitByBullet(), é o evento que verifica se levou algum tiro, se levar ele recua 10:

```
public void onHitByBullet(HitByBulletEvent e) {  
    back(10);  
}
```

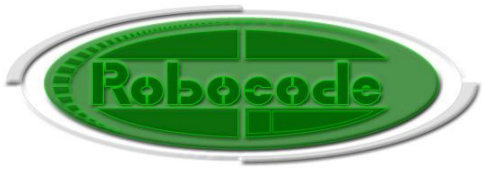


Programando Eventos do Robô

- **Evento onHitWall()**

- O Evento onHitWall(), é o evento que verifica se levou bateu na parede, se bater ele recua 10:

```
public void onHitWall(HitWallEvent e) {  
    back(20);  
}
```



Programando Eventos do Robô

- **Evento onHitRobot()**

- O Evento onHitRobot(), é o evento que verifica bateu em outro robô, se bater ele envia uma mensagem:

```
public void onHitRobot(HitRobotEvent e) {  
    System.out.println("Choquei com outro Robô");  
}
```

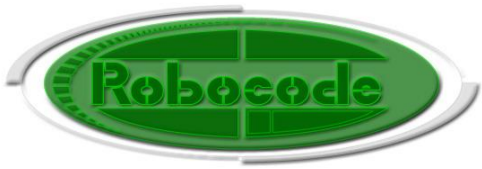


Programando Eventos do Robô

- **Evento onBattleEnded()**

- O Evento onBattleEnded(), é o evento que verifica se a batalha chegou ao fim, se chegar ele envia uma mensagem:

```
public void onBattleEnded(BattleEndedEvent e) {  
    System.out.println("Acabou");  
}
```



Programando Eventos do Robô

- **Evento onBulletHitBullet()**

- O Evento onBulletHitBullet(), é o evento que verifica se o tiro dado acertou outro tiro, se acertou ele envia uma mensagem:

```
public void onBulletHitBullet(BulletHitBulletEvent e) {  
    System.out.println("Acertei outro tiro");  
}
```

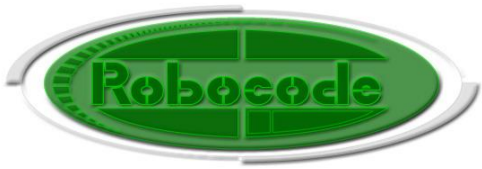
CONSTRUIR JUNTOS

Programando Eventos do Robô

- **Evento onBulletHit()**

- O Evento onBulletHit(), é o evento que verifica se o tiro dado acertou o alvo, se acertou ele envia uma mensagem

```
public void onBulletHit(BulletHitEvent e){  
    System.out.println("Acertei um tiro");  
}
```



Programando Eventos do Robô

- **Evento onBulletMissed()**

- O Evento onBulletMissed(), é o evento que verifica se o tiro dado errou o alvo, se errou ele envia uma mensagem

```
public void onBulletMissed(BulletMissedEvent e) {  
    System.out.println("Errei o tiro");  
}
```

Programando Eventos do Robô

- **Evento onDeath()**

- O Evento onDeath(), é o evento que verifica se acabou a partida para o nosso robô, se acabou ele envia uma mensagem

```
public void onDeath(DeathEvent e) {  
    System.out.println("Morri");  
}
```


Programando Eventos do Robô

- **Evento onRobotDeath()**

- O Evento onRobotDeath(), é o evento que verifica algum concorrente morreu, se morreu ele envia uma mensagem

```
public void onRobotDeath(RobotDeathEvent e) {  
    System.out.println("Um concorrente meu morreu");  
}
```

CONSTRUIR JUNTOS



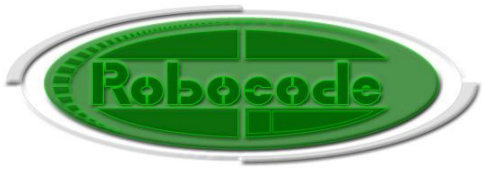
Programando Eventos do Robô

- **Evento onRoundEnded()**

- O Evento onRoundEnded(), é o evento que verifica se o round acabou, se acabou ele envia uma mensagem

```
public void onRoundEnded(RoundEndedEvent e) {  
    System.out.println("O Round Acabou agora");  
}
```

CONSTRUIR JUNTOS



Programando Eventos do Robô

- **Evento onWin()**

- O Evento onWin(), é o evento que verifica se o robô ganhou

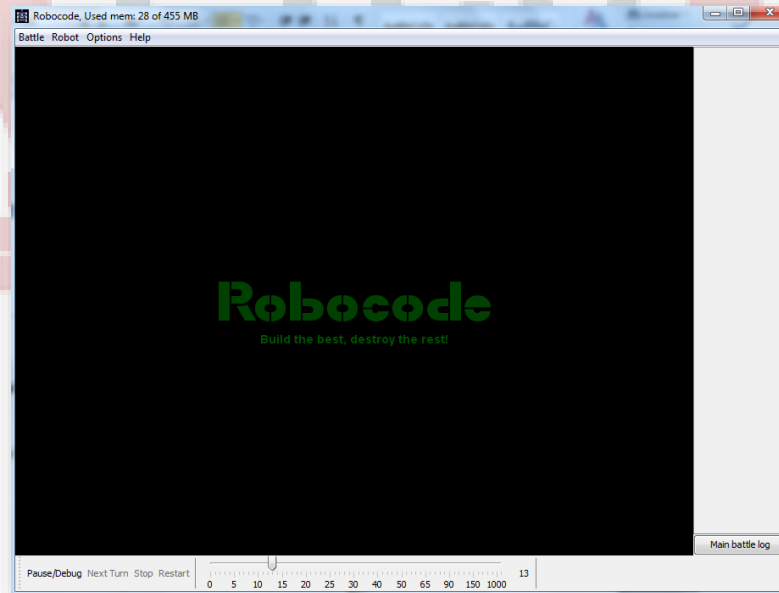
```
public void onWin(WinEvent e) {  
    System.out.println("Ganhei");  
}
```

CONSTRUIR JUNTOS



Empacotando o Robô

- A criação do pacote do seu Robô, serve para enviar o mesmo a um campeonato, por exemplo, onde será importado na nova arena de combate.
- Teremos que acessar novamente tela principal da arena, como mostra abaixo.



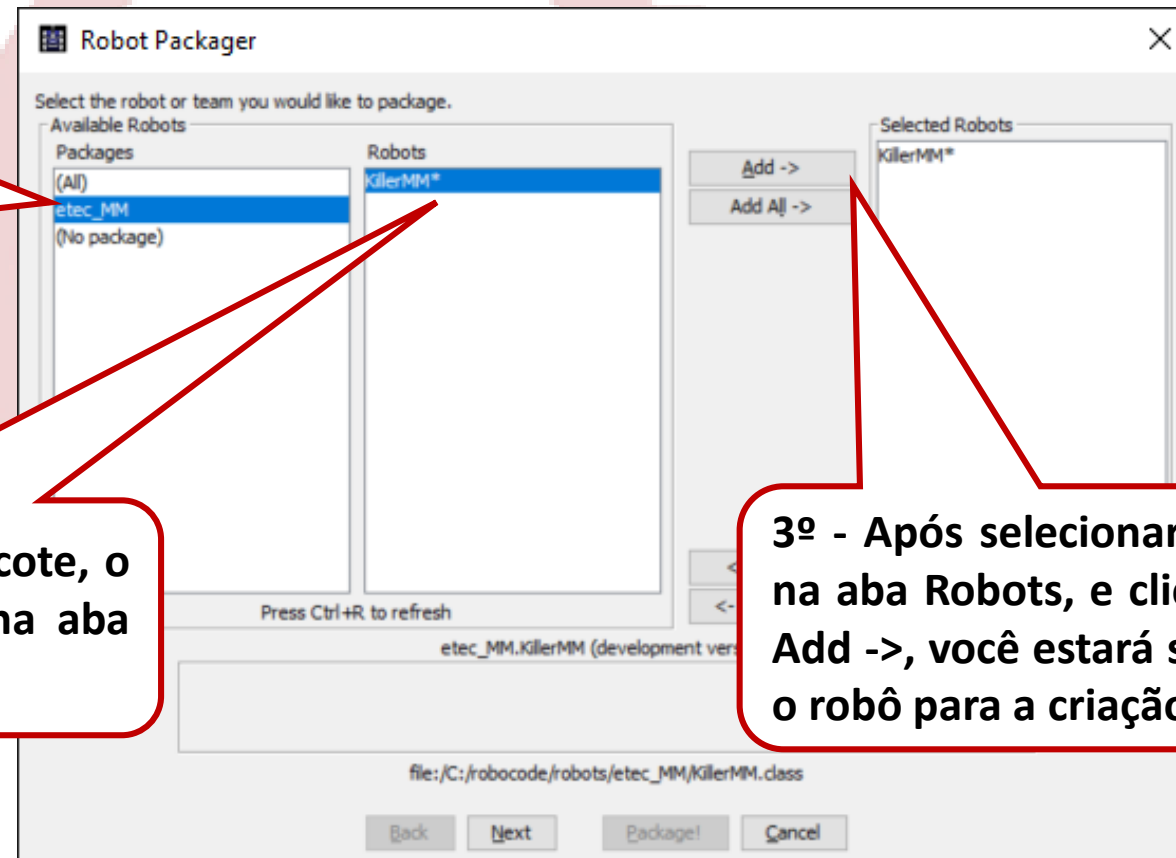
Empacotando o Robô

- Primeiro clicamos no menu **Robot** e em seguida na opção **Package robot or team** e abrirá a tela a seguir.

1º - Selecionar o pacote que se encontra o seu robô.

2º - Após a seleção do pacote, o robô criado será listado na aba Robots

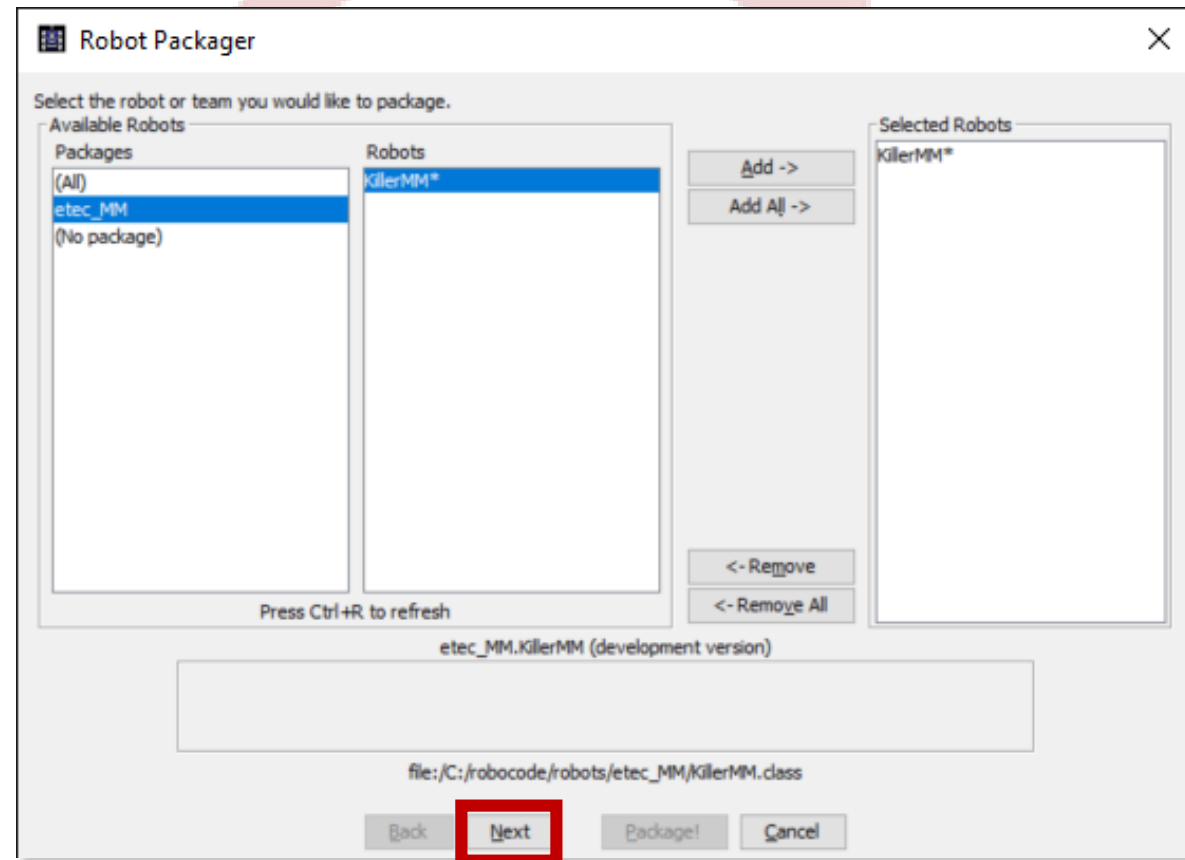
3º - Após selecionar o seu robô, na aba Robots, e clicar no botão Add ->, você estará selecionando o robô para a criação do pacote.





Empacotando o Robô

- Em seguida clicar no botão **Next**.

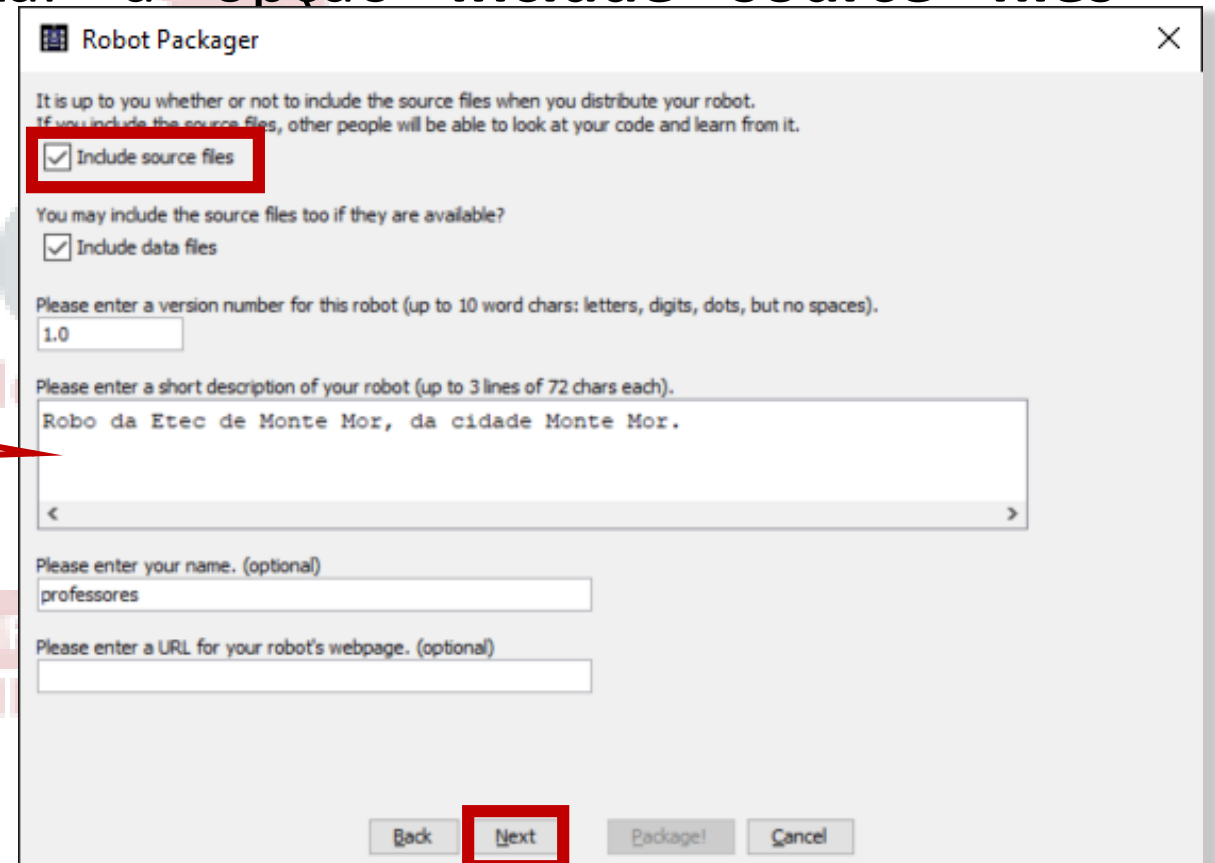


Empacotando o Robô

- Quando exibir esta tela, deixar a opção **Include source files** habilitado.

Inserir uma breve descrição a respeito do seu robô.

- Em seguida clicar no botão **Next**.



Robot Packager

It is up to you whether or not to include the source files when you distribute your robot.
If you include the source files, other people will be able to look at your code and learn from it.

Include source files

You may include the source files too if they are available?

Include data files

Please enter a version number for this robot (up to 10 word chars: letters, digits, dots, but no spaces).
1.0

Please enter a short description of your robot (up to 3 lines of 72 chars each).
Robo da Etec de Monte Mor, da cidade Monte Mor.

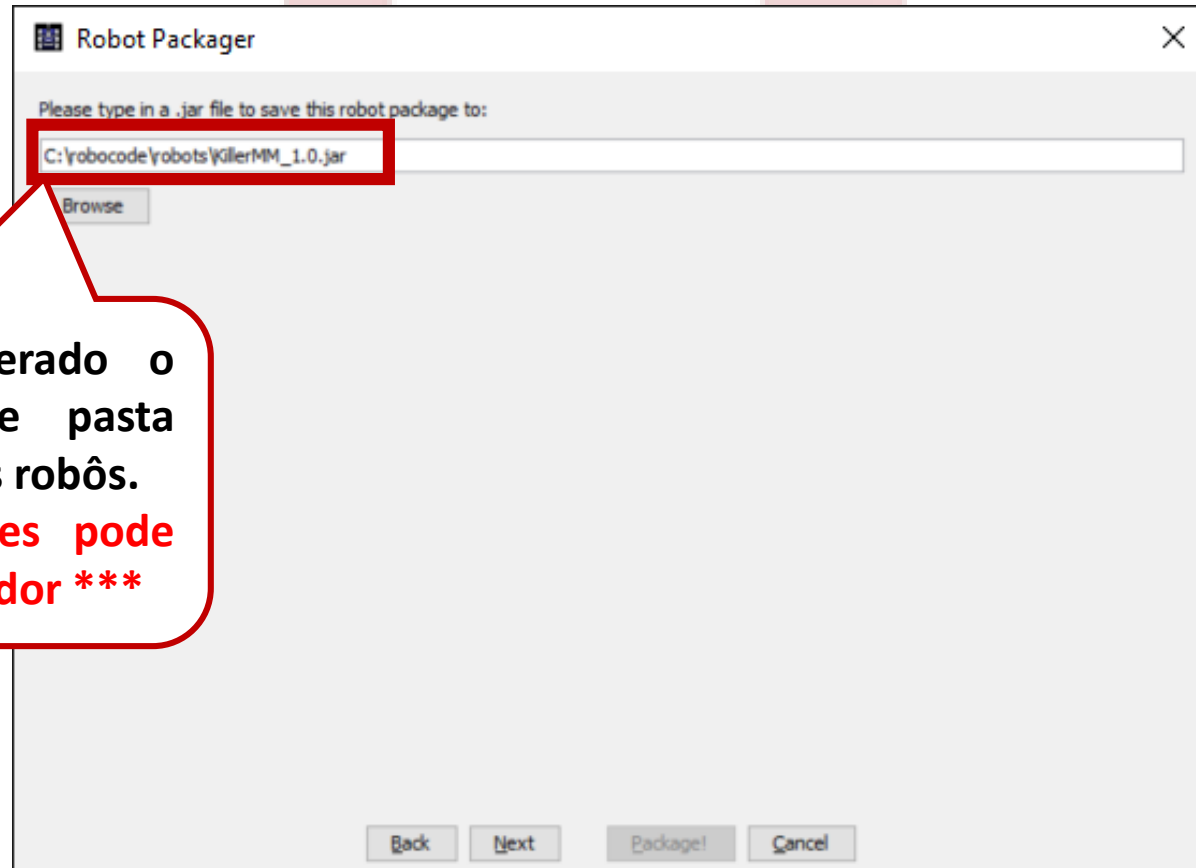
Please enter your name. (optional)
professores

Please enter a URL for your robot's webpage. (optional)

Back Next Package! Cancel

Empacotando o Robô

- Nesta tela informaremos o nome do pacote (.jar) a ser gerado.

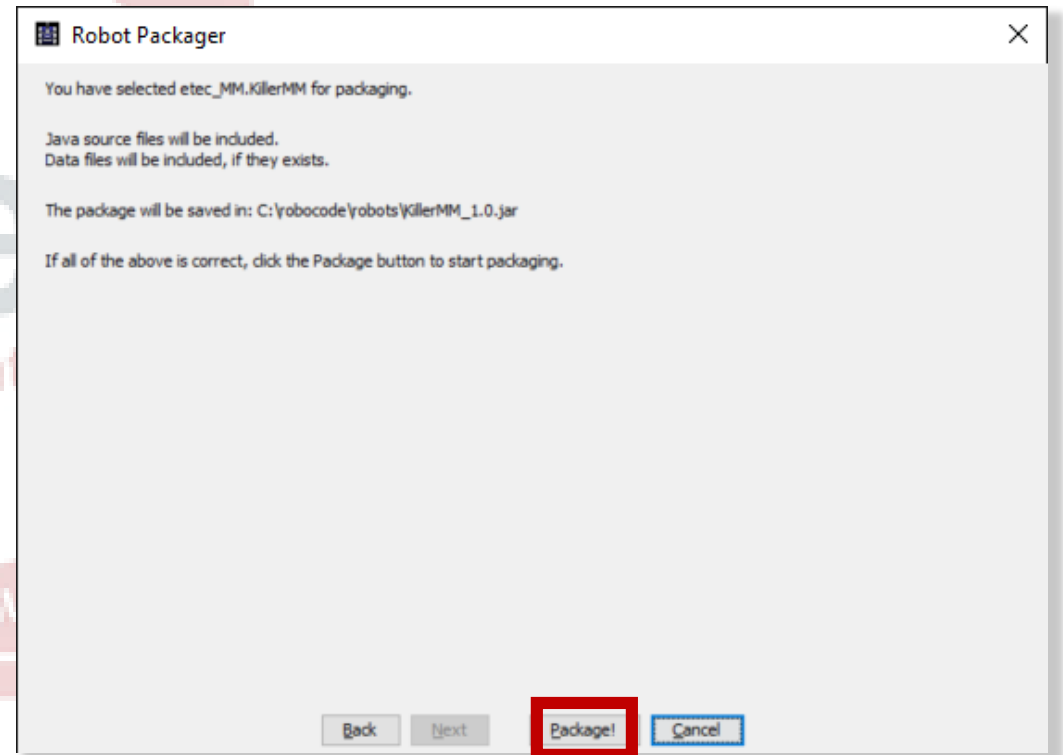


Notem que foi alterado o nome do pacote e pasta continua a mesma dos robôs.

***** Estas informações pode ser alteradas pelo criador *****

Empacotando o Robô

- Nesta tela será exibida as informações do robô selecionado, no qual será gerado o pacote.



- Agora é só clicar no botão **Package**.



Empacotando o Robô

- Esta tela mostra o processo do empacotamento do robô.

A screenshot of a 'Packaging results' dialog box. The dialog has a title bar with a close button (X) and an 'Edit' button. The main area contains the following text:

```
etec_MM/KillerMM.java  
etec_MM/KillerMM.properties  
etec_MM/KillerMM.class  
  
---- Codesize ----  
Codesize: 58 bytes  
Robot weight class: NanoBot (codesize < 250 bytes)
```

An 'OK' button is located at the bottom center of the dialog.

Empacotando o Robô

- Pacote do robô gerado. Este arquivo pode ser enviado para qualquer competição .

