Universidade do Minho
Escola de Engenharia
Departamento de Informática

Circumnavigation & Standard Odometer

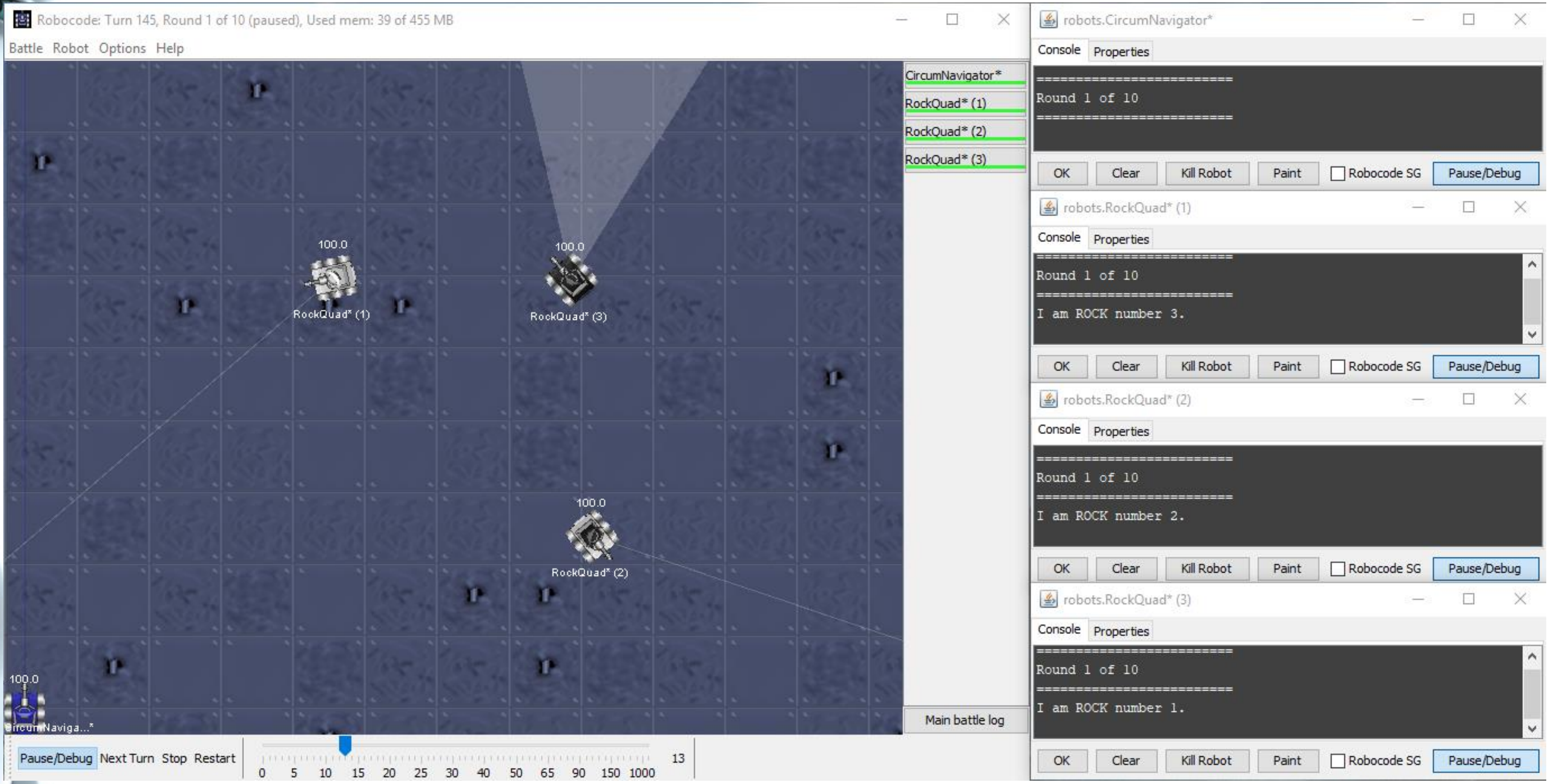
Sistemas Autónomos

Perfil Sistemas Inteligentes @ MEI/MiEI 1º/4º – 2º semestre

Cesar Analide, Bruno Fernandes

ISLab
Synthetic Intelligence Lab

Importing RockQuad to Robocode



robots.CircumNavigator*

```
=========================
Round 1 of 10
=========================
```

robots.RockQuad* (1)

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

robots.RockQuad* (2)

```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

robots.RockQuad* (3)

```
=========================
Round 1 of 10
=========================
I am ROCK number 1.
```

Importing RockQuad to Robocode

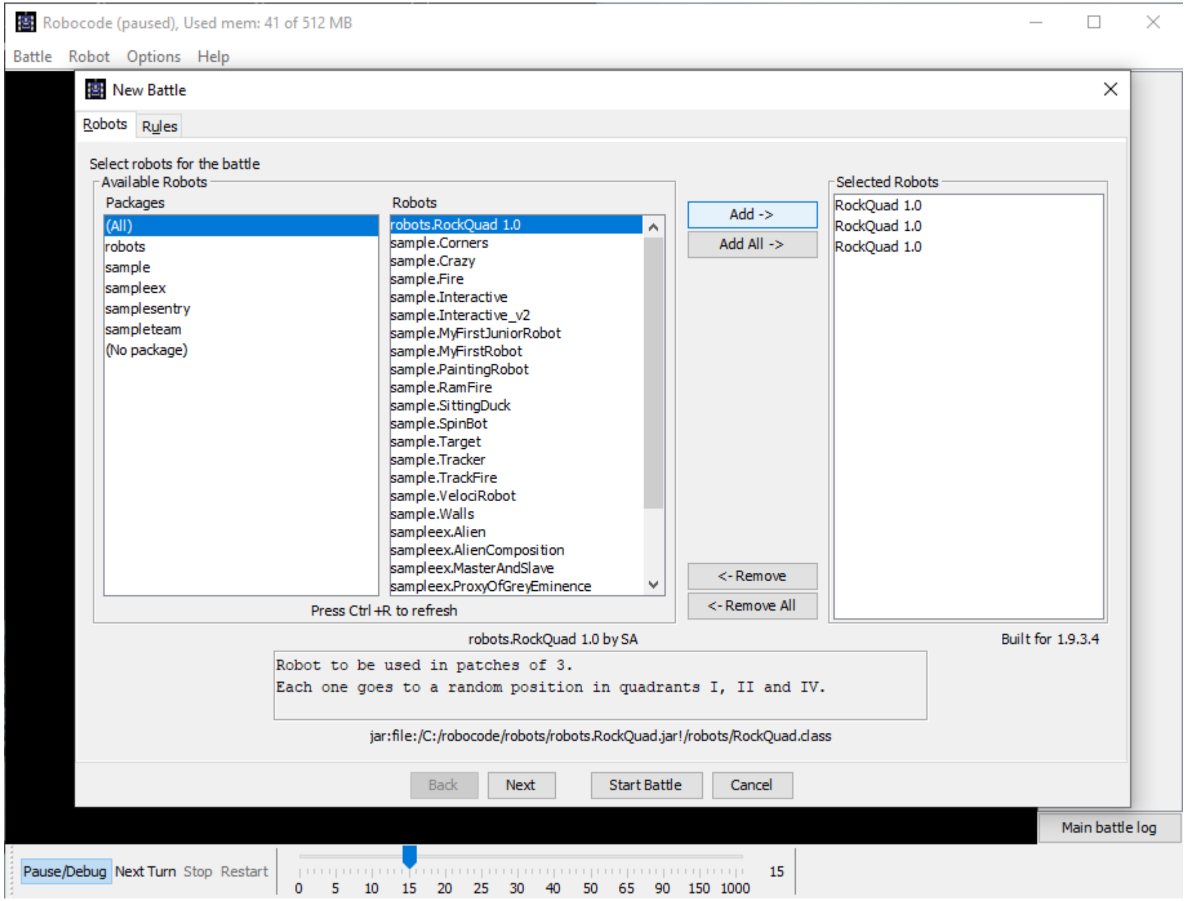
- Download RockQuad robot
  - https://goo.gl/PED2XF

- Import it in Robocode
  - Robot > Import robot or team > robots.RockQuad.jar
  - A new robot, entitled as RockQuad 1.0, should now be available

- To start a battle you must use 3 RockQuads (each one goes to a different quadrant, namely, quadrant I, II e IV) plus 1 robot (implemented by you) to go around the RockQuads!

ISLab
Synthetic Intelligence Lab

Let’s battle!



Select robots for the battle

Available Robots

Packages:
(All)
robots
sample
sampleex
samplesentry
sampleteam
(No package)

Robots:
robots.RockQuad 1.0
sample.Corners
sample.Crazy
sample.Fire
sample.Interactive
sample.Interactive_v2
sample.MyFirstJuniorRobot
sample.MyFirstRobot
sample.PaintingRobot
sample.RamFire
sample.SittingDuck
sample.SpinBot
sample.Target
sample.Tracker
sample.TrackFire
sample.VelociRobot
sample.Walls
sampleex.Alien
sampleex.AlienComposition
sampleex.MasterAndSlave
sampleex.ProxyOfGreyEminence

Press Ctrl +R to refresh

Selected Robots:
RockQuad 1.0
RockQuad 1.0
RockQuad 1.0

robots.RockQuad 1.0 by SA

Built for 1.9.3.4

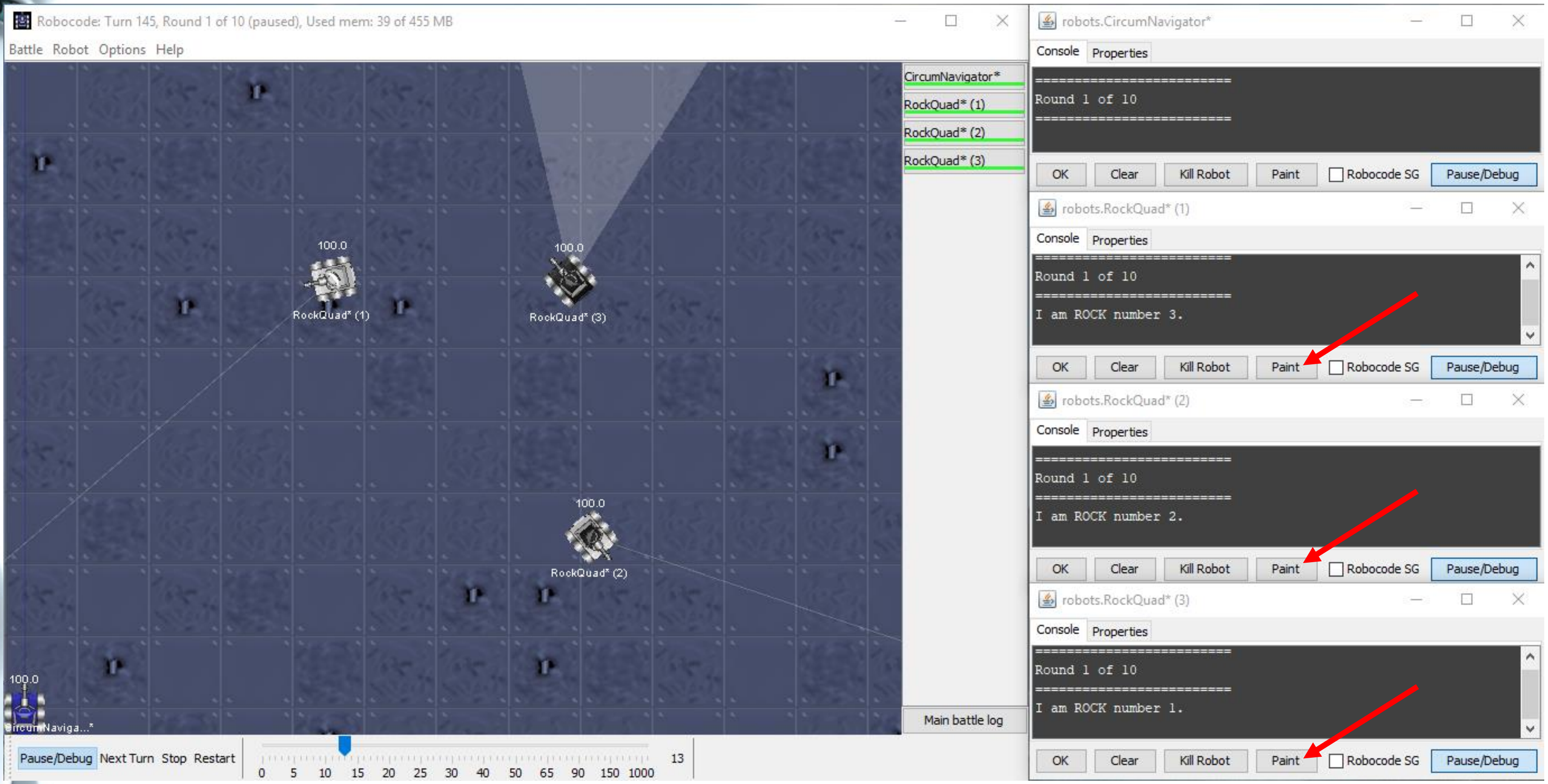
```
Robot to be used in patches of 3.
Each one goes to a random position in quadrants I, II and IV.
```

jar:file:/C:/robocode/robots/robots.RockQuad.jar!/robots/RockQuad.class

Let’s battle!

- Each RockQuad will go to a different quadrant
- Each RockQuad paints a line, setting the limits of the polygon to go around (on each robot’s console, press Paint)
- The RockQuad that goes to the first quadrant will print the total perimeter of the polygon
- Your robot, expected to go around the RockQuads in the shortest possible distance

Let's battle!



robots.CircumNavigator*

```
=========================
Round 1 of 10
=========================
```

robots.RockQuad* (1)

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

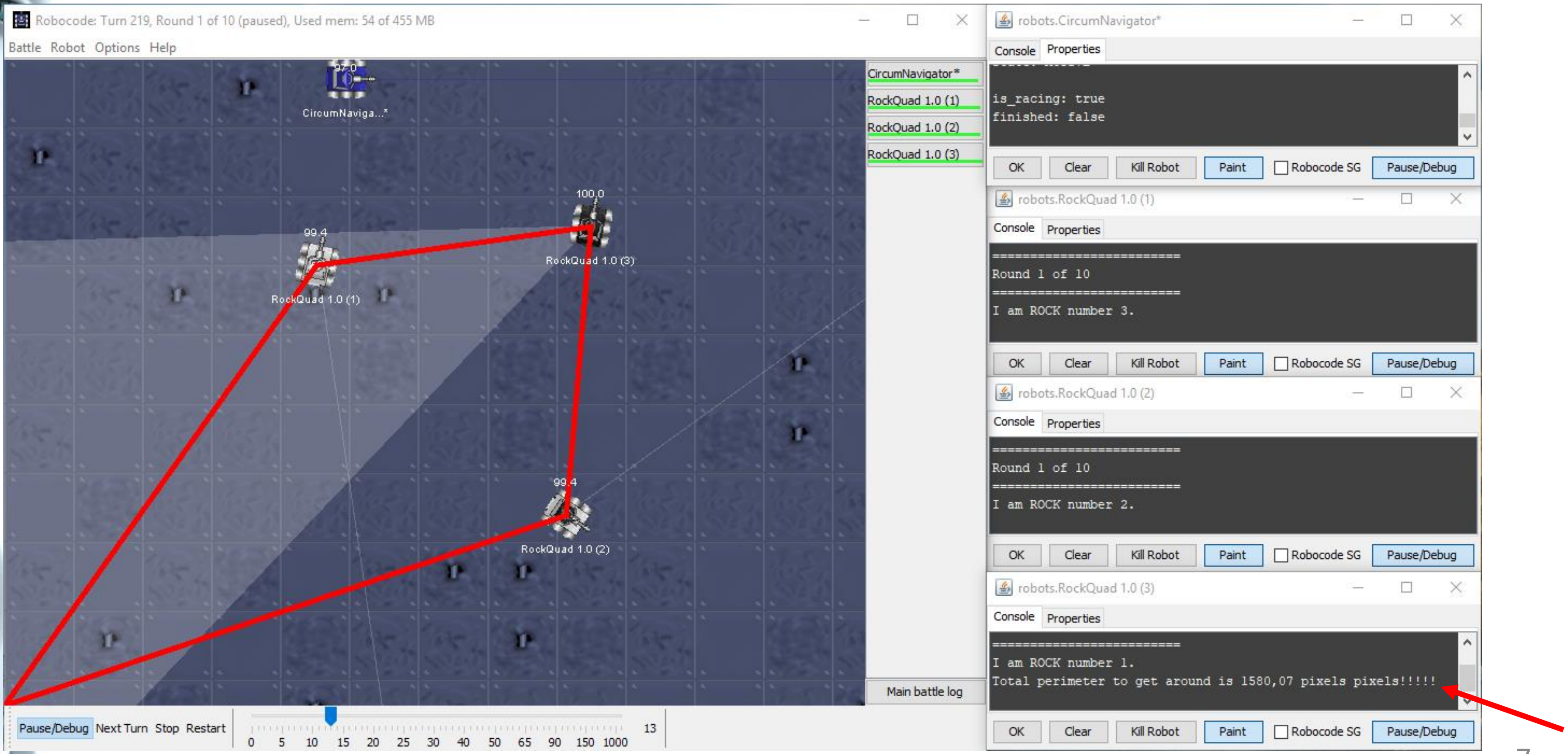
robots.RockQuad* (2)

```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

robots.RockQuad* (3)

```
=========================
Round 1 of 10
=========================
I am ROCK number 1.
```

Let's battle!



robots.CircumNavigator* – Console:

```
is_racing: true
finished: false
```

robots.RockQuad 1.0 (1) – Console:

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

robots.RockQuad 1.0 (2) – Console:

```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

robots.RockQuad 1.0 (3) – Console:

```
=========================
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!!
```

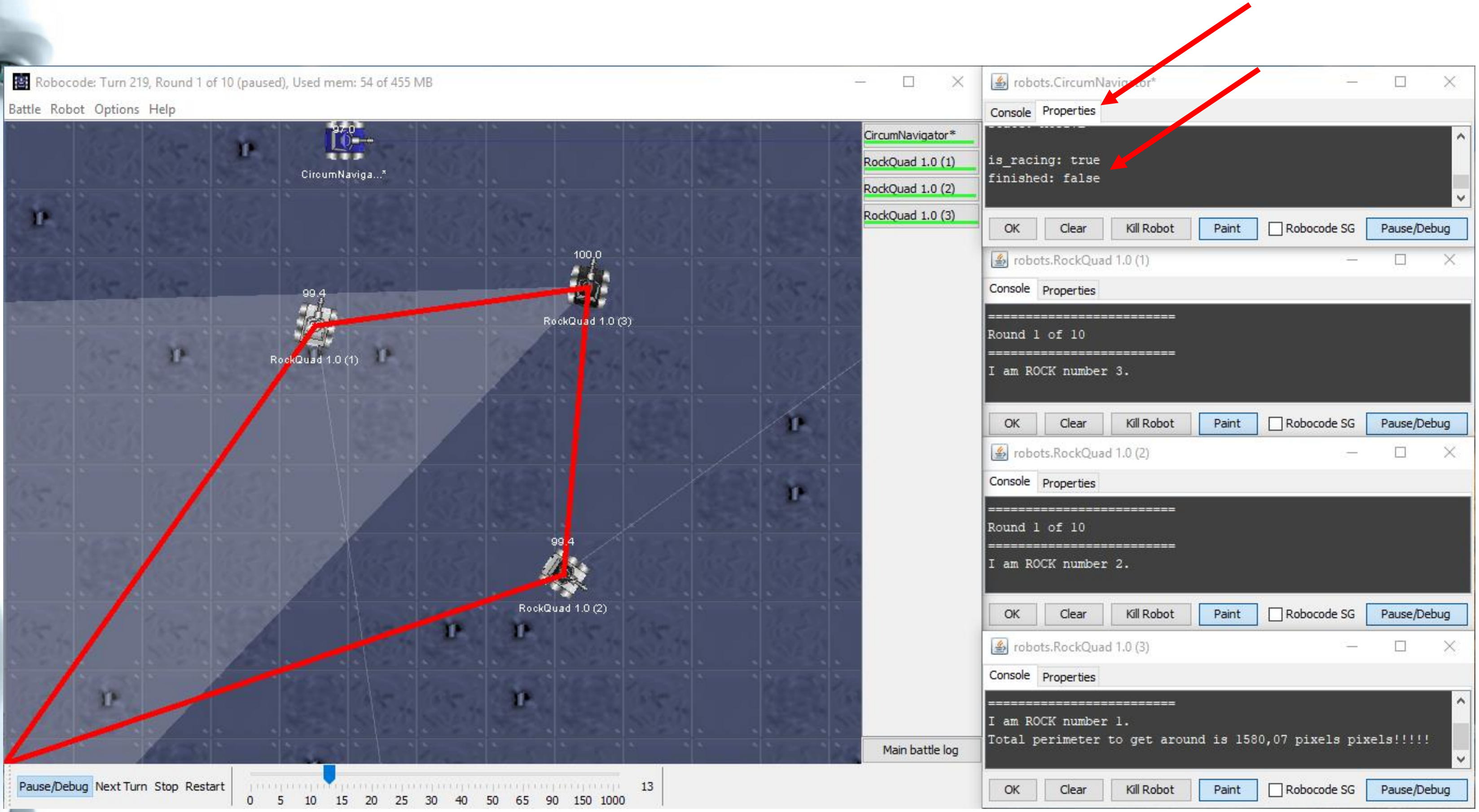
Using the StandardOdometer!

- Your robot, expected to go around the RockQuads in the shortest possible distance, will have a new set of personal properties as soon as you use the **StandardOdometer**:
  - is_racing - reveals if the robot is racing
  - finished - reveals if the race is finished

- The race starts as soon as your robot gets to the starting position (18, 18) and finishes as soon as it arrives to that same position

- As soon as the race is finished, the properties of your robot are updated and a new one emerges, race_distance, indicating the distance it took your robot to go around the RockQuads!

Using the StandardOdometer!



```
is_racing: true
finished: false
```

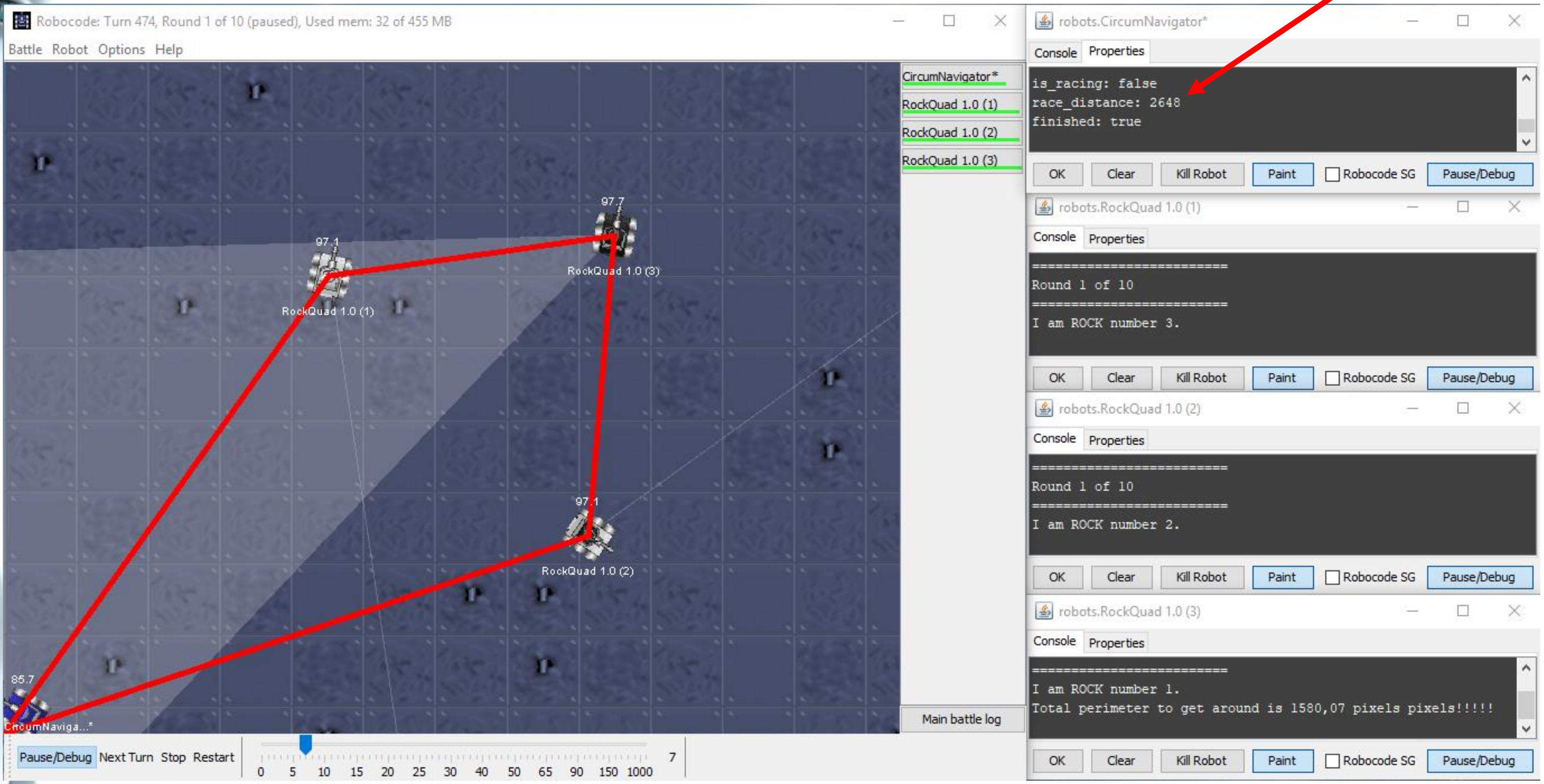
```
========================
Round 1 of 10
========================
I am ROCK number 3.
```

```
========================
Round 1 of 10
========================
I am ROCK number 2.
```

```
========================
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!!
```

ISLab
Synthetic Intelligence Lab

Using the StandardOdometer!



robots.CircumNavigator*

```
is_racing: false
race_distance: 2648
finished: true
```

robots.RockQuad 1.0 (1)

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

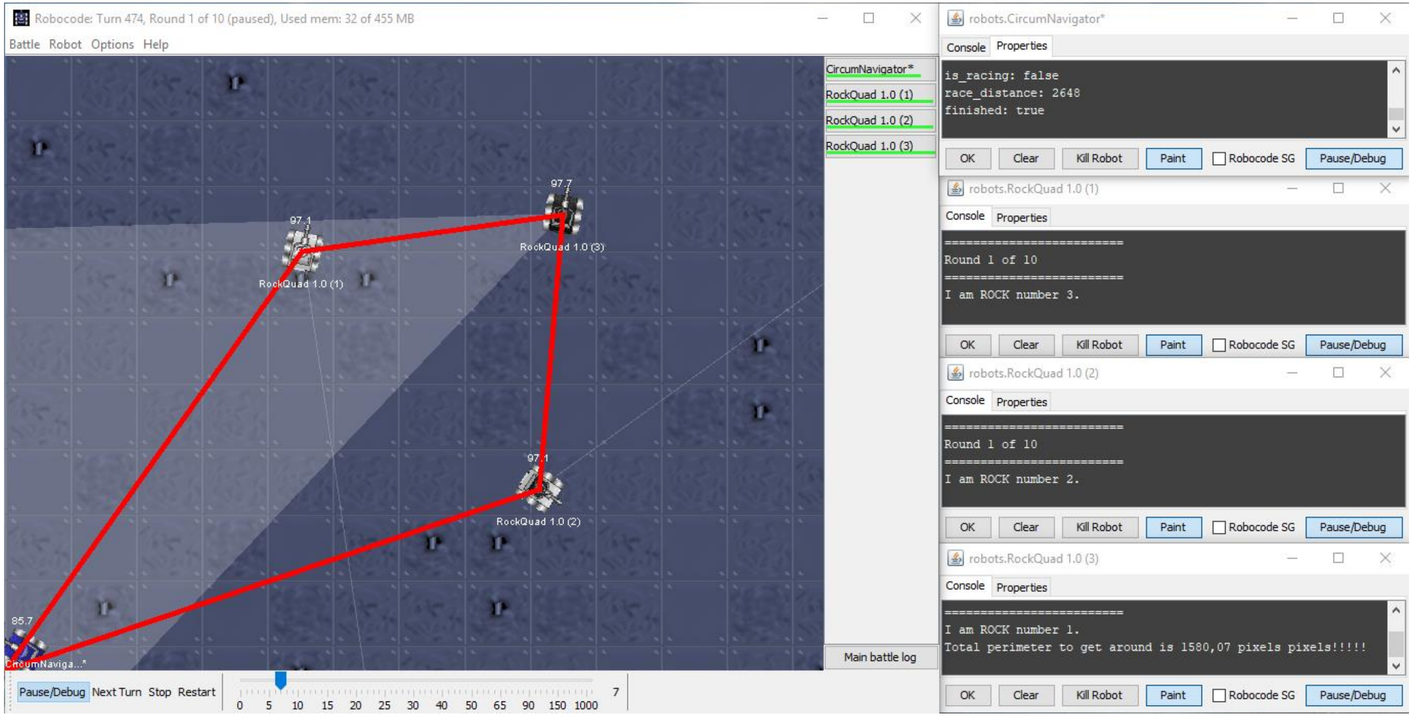
robots.RockQuad 1.0 (2)

```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

robots.RockQuad 1.0 (3)

```
=========================
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!!
```

Using the StandardOdometer!



```
is_racing: false
race_distance: 2648
finished: true
```

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

```
=========================
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!!
```

The goal is to have the best possible ratio (in this case, 59.7%):

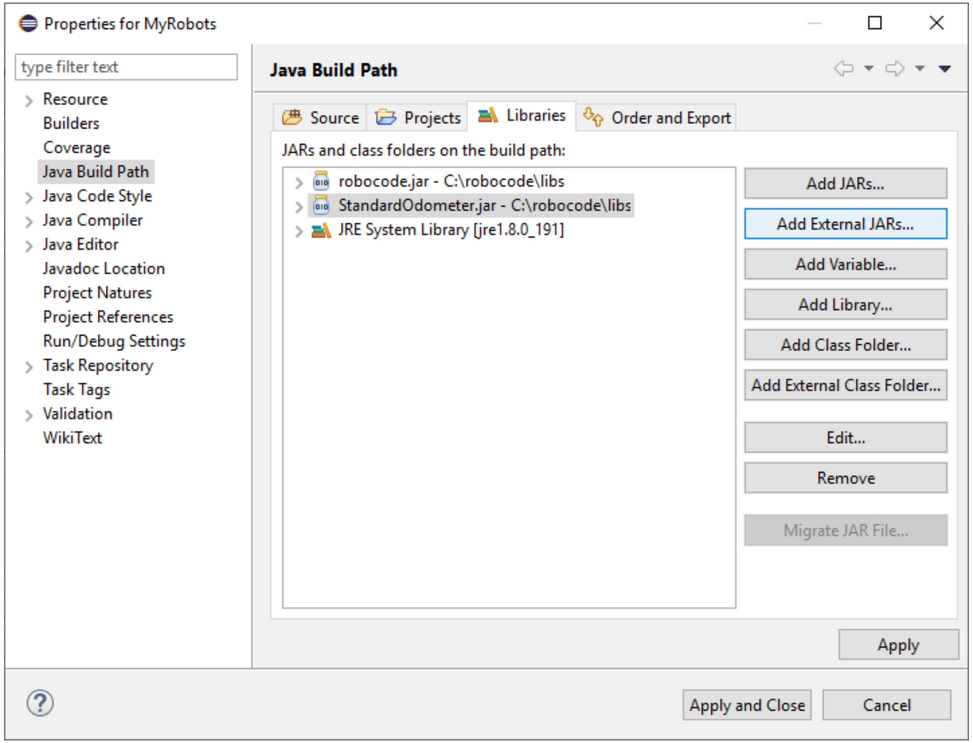
polygon_perimeter/race_distance

Importing the StandardOdometer to Robocode

- Download StandardOdometer.jar
  - https://goo.gl/jun6Sr
- Place the file in Robocode libs dir
  - Usually at C:\robocode\libs
- Go back a folder (C:\robocode) and edit the file entitled as robocode.bat
  - Change from: java -Xmx512M -cp libs/robocode.jar -XX:...
  - To (windows): java -Xmx512M -cp libs/robocode.jar;libs/StandardOdometer.jar; -XX:...
  - To (mac/linux): java -Xmx512M -cp libs/robocode.jar:libs/StandardOdometer.jar -XX:...

Adding the StandardOdometer to your project

- Add the StandardOdometer.jar to your project's buildpath so that your robots can use it

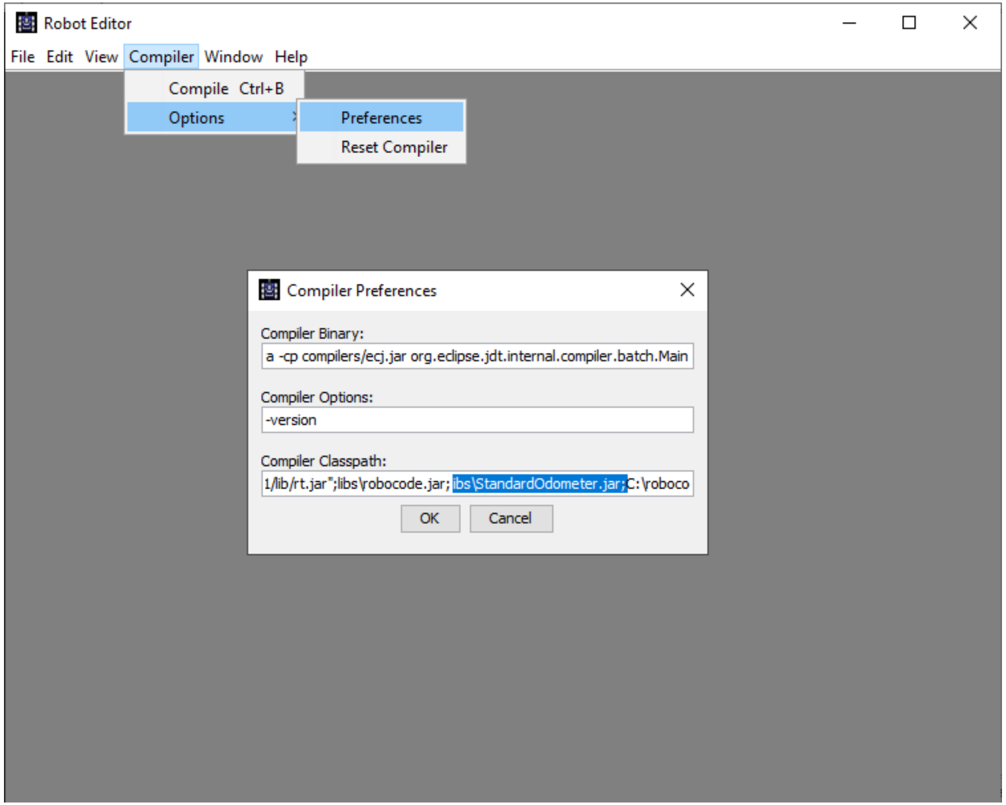


JARs and class folders on the build path:

robocode.jar - C:\robocode\libs
StandardOdometer.jar - C:\robocode\libs
JRE System Library [jre1.8.0_191]

Adding the StandardOdometer to your project

- Or, if you are using Robocode Source Editor, add it to the Compiler Classpath at Compiler > Options > Preferences



Compiler Binary:
a -cp compilers/ecj.jar org.eclipse.jdt.internal.compiler.batch.Main

Compiler Options:
-version

Compiler Classpath:
1/lib/rt.jar";libs\robocode.jar;ibs\StandardOdometer.jar;C:\roboco

Using the StandardOdometer

- To use the lib just (you must use AdvancedRobot):

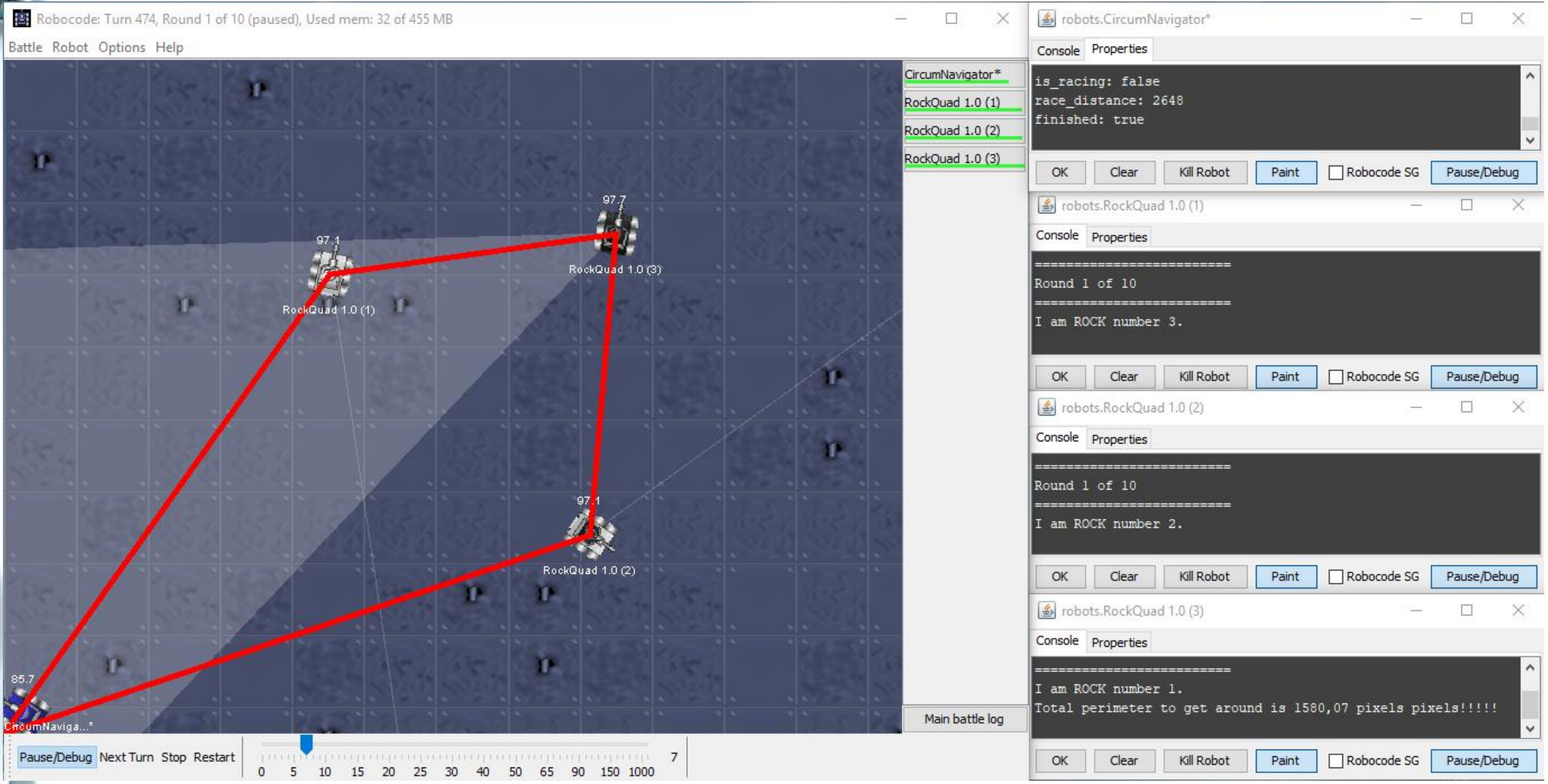
```
import standardOdometer.Odometer;
...

//** Private Instance Variable **\\
private Odometer odometer = new Odometer("IsRacing", this);
...

//** Add this inside run() **\\
addCustomEvent(odometer);
...

//** Method for handling the condition of race finished **\\
public void onCustomEvent(CustomEvent ev) {
    Condition cd = ev.getCondition();
    if (cd.getName().equals("IsRacing"))
        this.odometer.getRaceDistance();
}
```

Let's battle!



robots.CircumNavigator*

```
is_racing: false
race_distance: 2648
finished: true
```

robots.RockQuad 1.0 (1)

```
=========================
Round 1 of 10
=========================
I am ROCK number 3.
```

robots.RockQuad 1.0 (2)

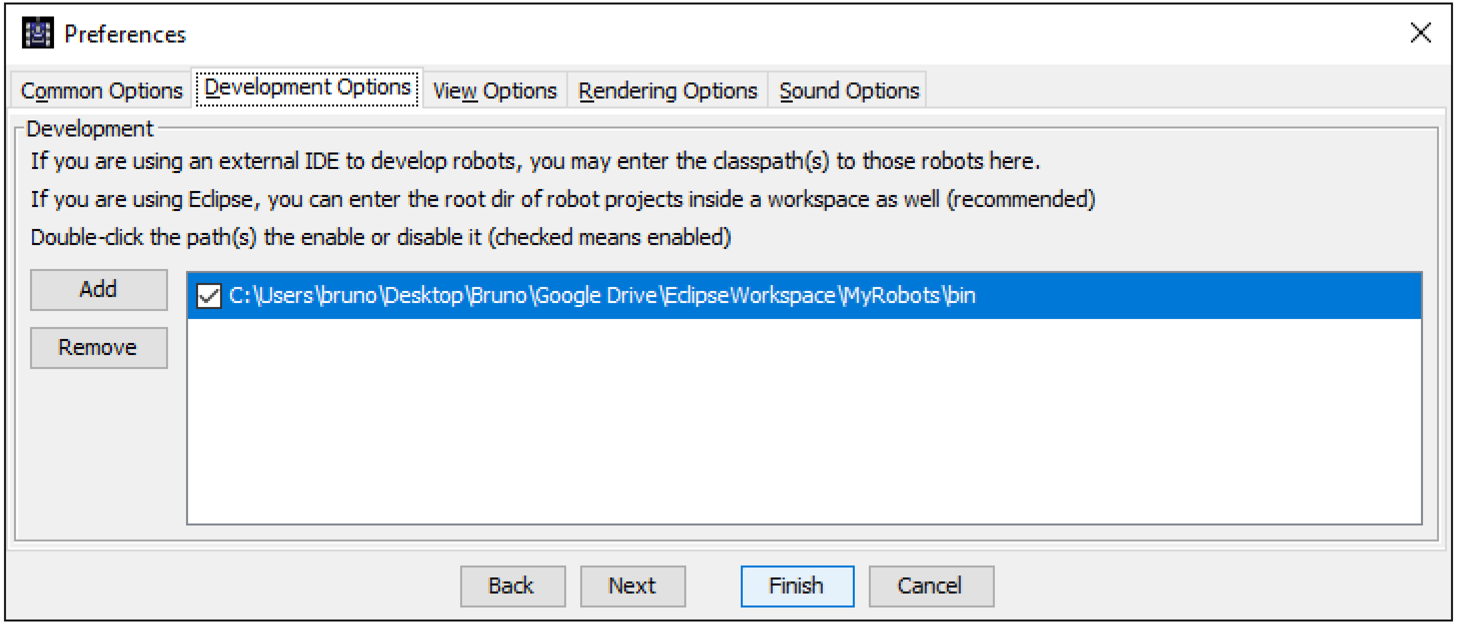
```
=========================
Round 1 of 10
=========================
I am ROCK number 2.
```

robots.RockQuad 1.0 (3)

```
=========================
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!!
```

Importing RockQuad to Robocode

- A small tip first... If you are using an IDE to develop the robots than make sure to add the classpath to the robots in robocode for them to be added automatically to the list of robots:



Development

If you are using an external IDE to develop robots, you may enter the classpath(s) to those robots here.

If you are using Eclipse, you can enter the root dir of robot projects inside a workspace as well (recommended)

Double-click the path(s) the enable or disable it (checked means enabled)

☑ C:\Users\bruno\Desktop\Bruno\Google Drive\EclipseWorkspace\MyRobots\bin