

Standard Odometer - How To

Sistemas Autónomos

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

Cesar Analide, Bruno Fernandes

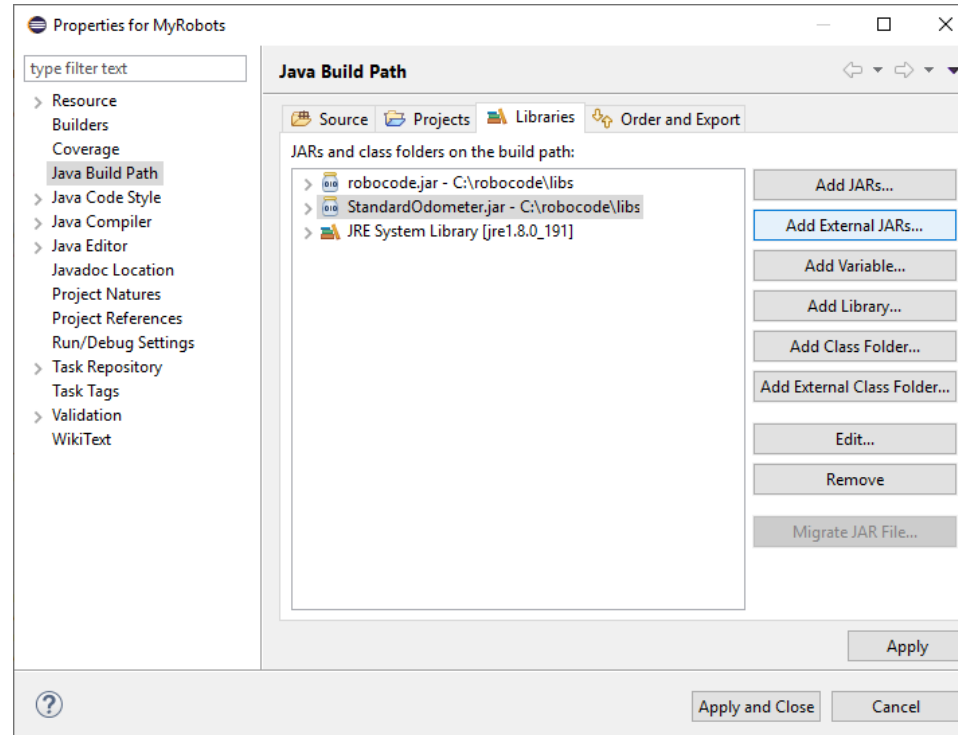


Importing the lib to Robocode

- 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: java -Xmx512M -cp **libs/robocode.jar;libs/StandardOdometer.jar**; -XX:...

Importing the lib to the project

- Add StandardOdometer.jar to your **project's buildpath**





- To use the lib just:

```
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();
}
```



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 to go around** the RockQuads



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 (1) **print the total perimeter of the polygon** and (2) will set the perimeter as a personal property
- Your robot, expected to go around the RockQuads in the shortest possible distance, will have a **new set of personal properties**:
 - `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 **amount of pixels it took your robot to go around** the RockQuads!

Let's battle!



Robocode: Turn 145, Round 1 of 10 (paused), Used mem: 39 of 455 MB

Battle Robot Options Help

100.0
RockQuad* (1)

100.0
RockQuad* (3)

100.0
RockQuad* (2)

100.0
CircumNavi...

Pause/Debug Next Turn Stop Restart

0 5 10 15 20 25 30 40 50 65 90 150 1000 13

Main battle log

robots.CircumNavigator*

Console Properties

Round 1 of 10

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad* (1)

Console Properties

Round 1 of 10

I am ROCK number 3.

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad* (2)

Console Properties

Round 1 of 10

I am ROCK number 2.

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad* (3)

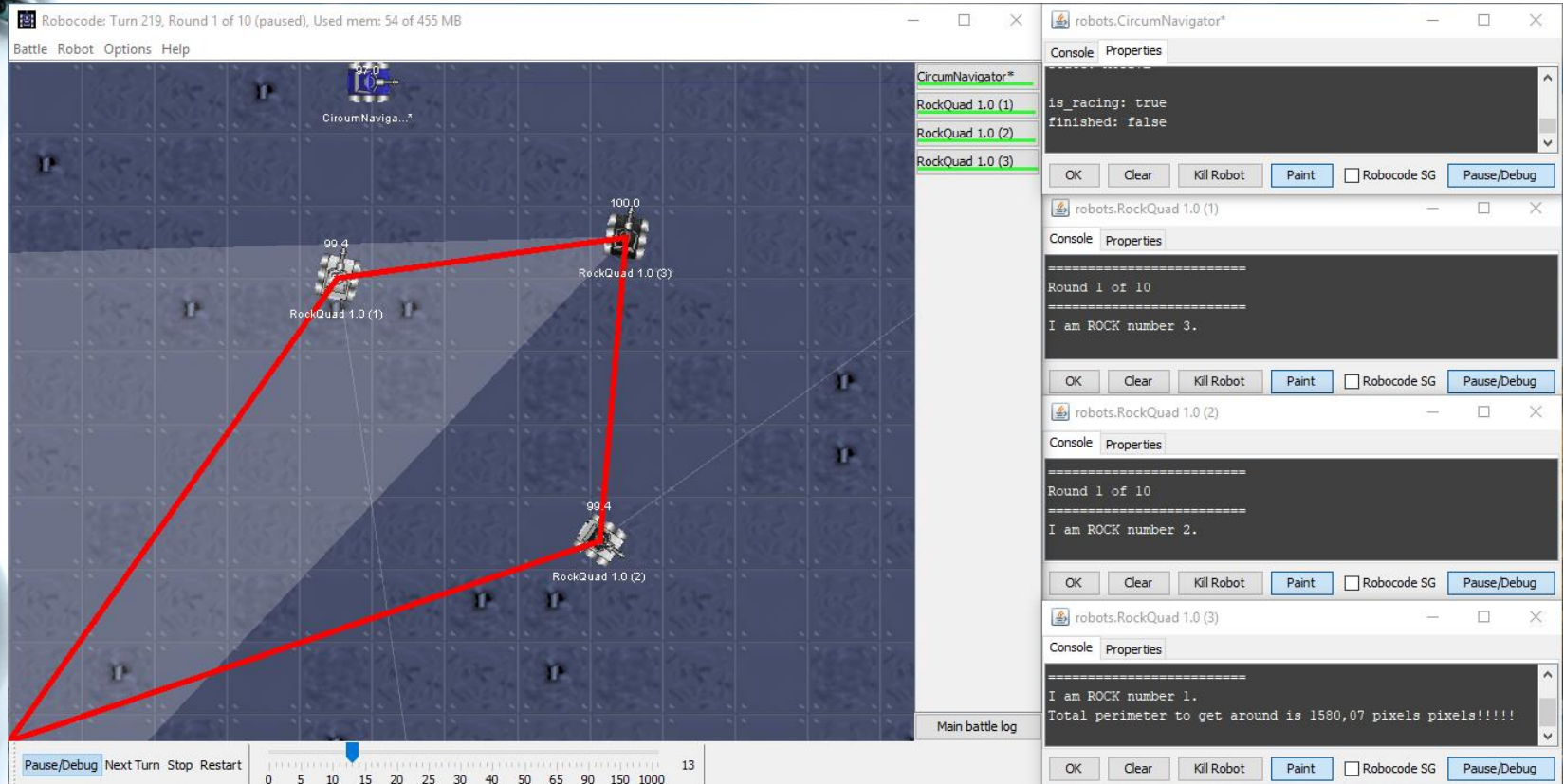
Console Properties

Round 1 of 10

I am ROCK number 1.

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

Let's battle!

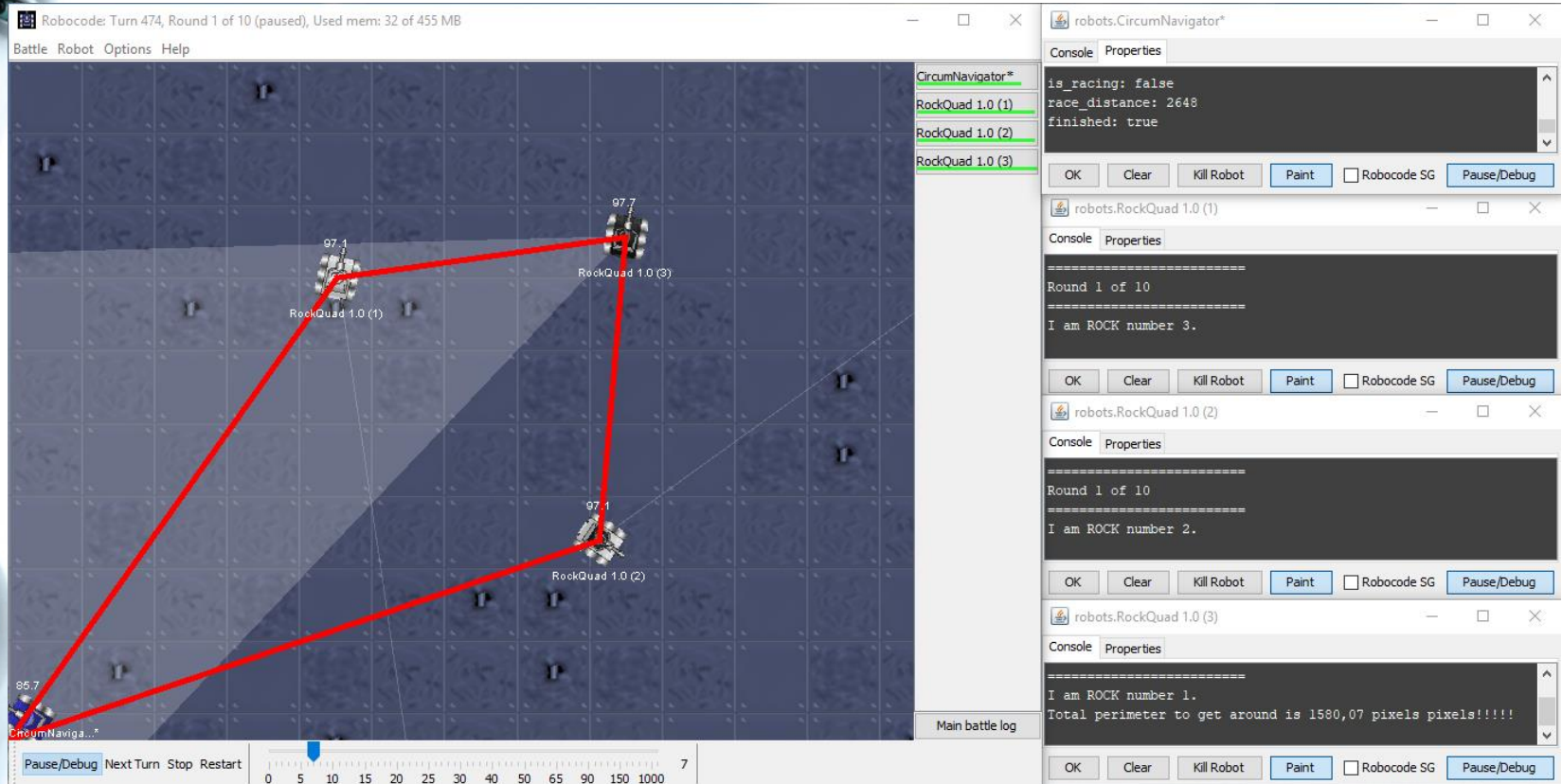


The image displays a Robocode battle arena and several console windows. The main arena window, titled "Robocode: Turn 219, Round 1 of 10 (paused), Used mem: 54 of 455 MB", shows a dark blue grid with a light blue diagonal line. Three robots are visible: "CircumNavigator*" at the top center, "RockQuad 1.0 (1)" at the bottom left, and "RockQuad 1.0 (2)" at the bottom right. Red lines connect the three robots, forming a triangle. The bottom of the arena window has a "Main battle log" button and a timeline from 0 to 13 seconds.

Three console windows are open on the right side of the arena:

- robots.CircumNavigator***
Console: is_racing: true, finished: false
Buttons: OK, Clear, Kill Robot, Paint, ☐ Robocode SG, Pause/Debug
- robots.RockQuad 1.0 (1)**
Console: Round 1 of 10, I am ROCK number 3.
Buttons: OK, Clear, Kill Robot, Paint, ☐ Robocode SG, Pause/Debug
- robots.RockQuad 1.0 (2)**
Console: Round 1 of 10, I am ROCK number 2.
Buttons: OK, Clear, Kill Robot, Paint, ☐ Robocode SG, Pause/Debug
- robots.RockQuad 1.0 (3)**
Console: Round 1 of 10, I am ROCK number 1. Total perimeter to get around is 1580,07 pixels pixels!!!!
Buttons: OK, Clear, Kill Robot, Paint, ☐ Robocode SG, Pause/Debug

Let's battle!



Robocode: Turn 474, Round 1 of 10 (paused), Used mem: 32 of 455 MB

Battle Robot Options Help

CircumNavigator*

RockQuad 1.0 (1)

RockQuad 1.0 (2)

RockQuad 1.0 (3)

Console Properties

is_racing: false
race_distance: 2648
finished: true

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.CircumNavigator*

Console Properties

Round 1 of 10
I am ROCK number 3.

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad 1.0 (1)

Console Properties

Round 1 of 10
I am ROCK number 2.

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad 1.0 (2)

Console Properties

Round 1 of 10
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

robots.RockQuad 1.0 (3)

Console Properties

Round 1 of 10
I am ROCK number 1.
Total perimeter to get around is 1580,07 pixels pixels!!!!

OK Clear Kill Robot Paint ☐ Robocode SG Pause/Debug

Main battle log

Pause/Debug Next Turn Stop Restart

0 5 10 15 20 25 30 40 50 65 90 150 1000 7