[(https://www.generationrobots.com/en/content/65-ultrasonic-sonar-sensors-for-robots)](https://www.generationrobots.com/en/content/65-ultrasonic-sonar-sensors-for-robots)

Ultrasonic Sensors

Ultrasonic sensors work by measuring the return time of a high-frequency sound wave emitted by the sensor (over 20,000 Hz, which is therefore inaudible to humans). As the speed of sound is essentially known, the obstacle's distance can then be deduced.

The distance d of the object causing the reflection is:

$$d = \frac{c \cdot t}{2}$$

where c is the speed of the sound (343 m/s in air at standard pressure and 20°C) and t is the time of flight.

The Pioneer 3-DX robot includes 8 forward-facing ultrasonic sensors, and 8 optional rear-facing sonar for distance measurements. In the simulations of this course, we are going to use only the forward-facing set of sensors.

In [1]:

```
import packages.initialization
import pioneer3dx as p3dx
p3dx.init()
```

With the following GUI widget you can move the robot around.

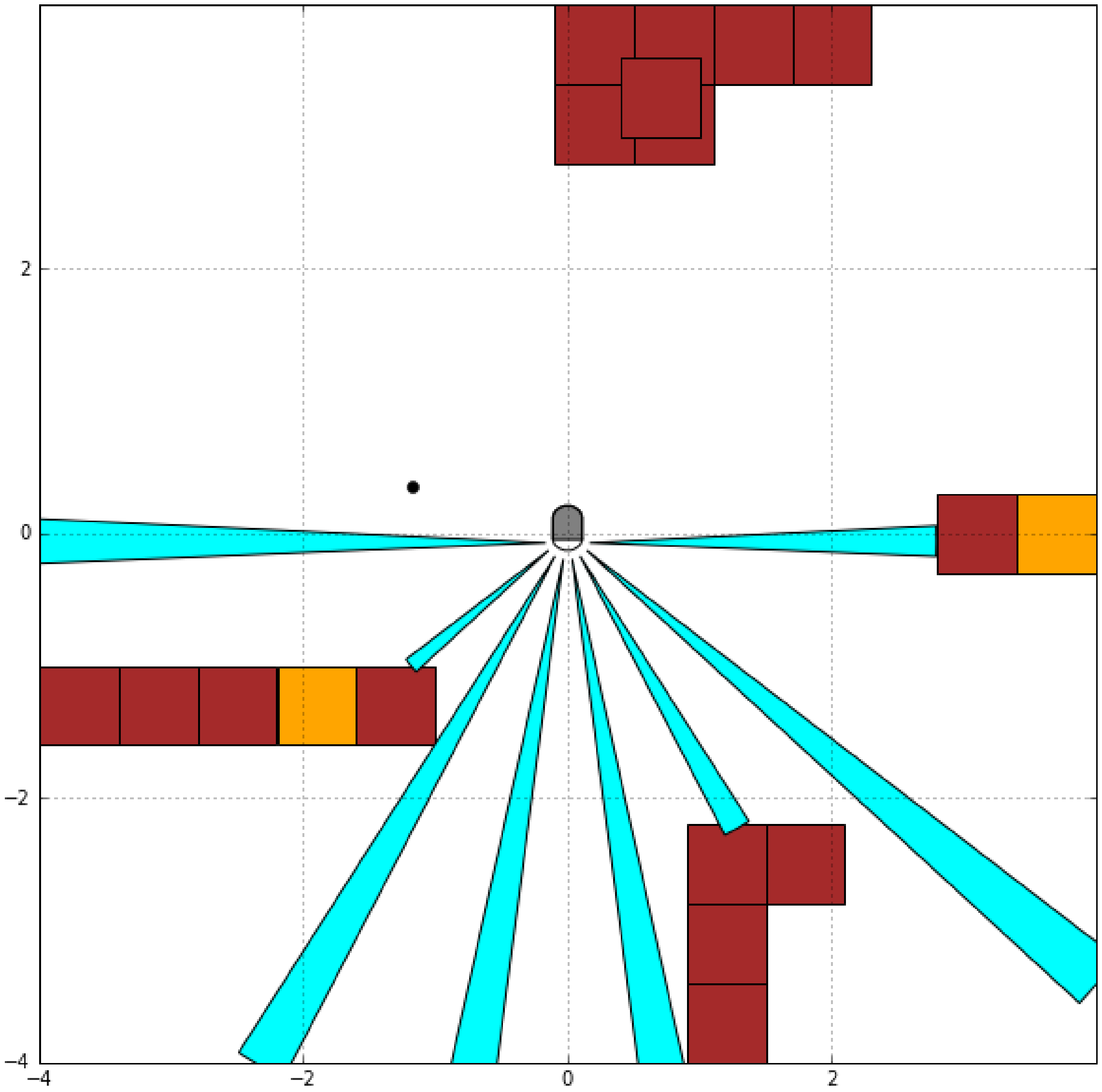
In [2]:

```
%matplotlib inline
import ultrasonic
import motion_widget
```

And you can plot a diagram with the position of the robot in the environment, and the measurements of the ultrasonic sensors.

In [3]:

```
ultrasonic.plot()
```



The values of the measurements are stored in an array:

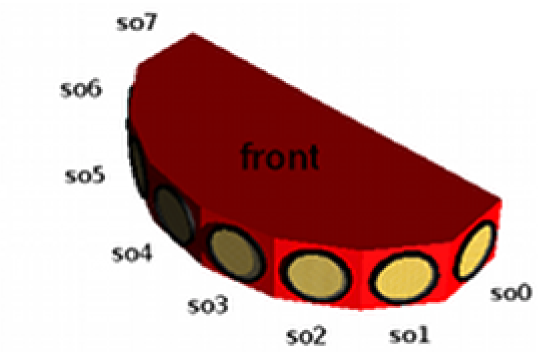
In [4]:

```
p3dx.distance
```

Out[4]:

```
[2.6657397747039795,
 5.108361721038818,
 2.3858048915863037,
 3.839787483215332,
 3.854698419570923,
 4.474814414978027,
 1.4037491083145142,
 3.8946311473846436]
```

(https://www.cyberbotics.com/doc/guide/using-the-pioneer-3-at-and-pioneer-3-dx-robots) The sensors are numbered from 0 to 7 starting from the left side of the robot, in clockwise order (see the figure).



Each measurement can be read individually with the appropriate index in the array, e.g. the first measurement is

```
p3dx.distance[0]
```

The following code uses a `for` loop for displaying the values of the eight sensors with a precision of three digits. Even if the robot is not moving, the values are likely to be different from those measured above, because of the (simulated) noise in the readings.

In [5]:

```
for i in range(8):
    print("so%d: %.3f m" % (i, p3dx.distance[i]))
```

```
so0: 2.654 m
so1: 5.062 m
so2: 2.347 m
so3: 3.890 m
so4: 3.902 m
so5: 4.503 m
so6: 1.413 m
so7: 3.853 m
```

Let's test the sensors in a first simple application: detecting obstacles (Detecting%20Obstacles.ipynb).

Try-a-Bot: an open source guide for robot programming

Developed by:

Universitat Jaume I Robotic Intelligence Lab

(http://robinlab.uji.es)

Sponsored by:

IEEE Robotics & Automation Society (http://www.ieee-ras.org)

CYBERBOTICS professional mobile robot simulation (http://www.cyberbotics.com)

The Construct Just Simulate (http://www.theconstructsim.com)

Follow us:

(https://www.facebook.com/RobotProgrammingNetwork) (https://www.youtube.com/user/robotprogrammin