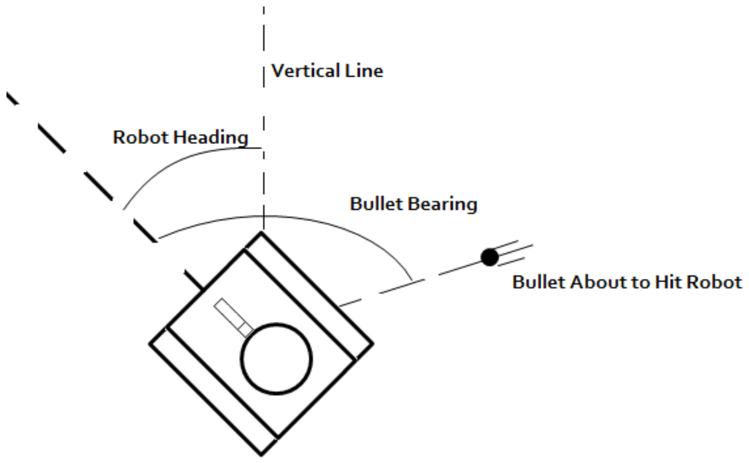
Robocode

Concepts and Quick Start Guide



A **bearing** is a relative angle in degrees to your robot's direction. A **heading** is the angle something makes with vertical. Most of the things your robot knows about the world will be in terms of bearings and headings. In both terms, clockwise is the positive direction.

A robot has **energy**. Energy starts at 100 and goes down whenever a robot fires, gets hit or runs full speed into a wall. Energy goes up when a bullet the robot fired hits a target.

Actions:

What to type	What it does
ahead(**distance**);	Makes the robot go ahead **distance** pixels. **Distance** can be a whole number, decimal number, positive or negative. Going ahead a negative number makes the robot go backwards.
back(**distance**);	Makes the robot go backwards **distance** pixels. If distance is negative the robot will go forward.
fire(**power**);	Fires a bullet with a given **power**. **Power** is a number 0.1 to 3. The power is subtracted from your energy, so don't blindly fire! The greater the power, the greater the damage when it hits.

turnRight(**angle**);
turnLeft(**angle**);
	Turns the robot (gun and all) left or right **angle** degrees. If **angle** is negative the robot will turn the other direction.

turnGunLeft(**angle**);
turnGunRight(**angle**);
	Turns only the gun left/right **angle** degrees. If **angle** is negatie the gun turns in the other direction.

When performing actions, we aren't constrained to predetermined numbers to turn and move. Any of these numbers can also be based on math, and other information that we can gather. Inside the parentheses we can use +-*/ digits and more parentheses to do math. Wherever a number is needed, we can either include it at the time we write the program, or use an information gathering function.

Information Gathering:

What to type	What it will give us
getHeading()	The heading of our robot. Will be in the range 0-360.
getGunHeading()	The heading of our robot's gun. Will be in the range 0-360.
getX()	The x position of our robot. 0 is the left side of the battlefield.
getY()	The y position of our robot. 0 is the bottom of the battlefield.
getBattleFieldHeight()	The width/height of the battlefield.

getBattleFieldWidth()	

Scanning and Events:

Whenever the robot moves or turns, or when the gun turns, the radar attached to the gun will be "scanning." When a robot is "seen" by this scanner it will trigger an onScannedRobot **event**. Events are given to small bits of code that should respond to them. In addition to knowing that an event has happened, we can be given additional information. In the events this special information can be retrieved in the following ways:

onScannedRobot

What to type	What you get
event.getBearing()	The **bearing** between you and the robot you scanned.
event.getDistance()	The center to center distance between you and the robot you scanned.

onHitByBullet

What to type	What you get
event.getBearing()	The relative angle the bullet hit you.

Making Decisions:

Our robot will often need to decide for itself what is a good idea and a bad idea. One common decision to make is to fire only when we can "see the whites of their eyes" when we scan a robot. In an *if statement* conditions are made by inequalities, and the actions are only done if the conditions are met. An example if statement would be:

```
if(event.getDistance() <=300) {
        fire(1);
}
```

This code snippet if placed inside onScannedRobot would fire lightly at the scanned robot if it was less than 300 units away. In between the parentheses, put what the condition is to be tested. In the place of fire(1); put what you want to happen if the condition is met.