AIBO Movement Control

The AIBO can be moved using the standard forward/backward and rotate interfaces: `robot.move(`*translate*`, `*rotate*`)`. There is also an additional control: `robot.strafe(`*amount*`)`. Strafing is a side-to-side movement.

Three different types of walks can be imported using the command `robot.setWalk(`*string*`)`, where *string* is one of the following: "pace", "tiger", "walk"

Individual Joint Control

The AIBO has a number of joints to be manipulated. The values of the joint positions are normalized to a range from -1.0 to 1.0, or in some cases 0.0 to 1.0. This table lists the joint names, ranges, and some useful values to know.

Joint	**Range**	**Significant values**
mouth	0.0 : 1.0	0.0 is closed, 1.0 is open
tail pan	-1.0 : 1.0	
tail tilt	-1.0 : 1.0	
front rotator	-1.0 : 1.0	0.0 pointing down, 1.0 max forward, -1.0 max backward
back rotator	-1.0 : 1.0	0.0 pointing down, 1.0 max backward, -1.0 max forward
elevator	-1.0 : 1.0	0.0 straight down, 1.0 max away from body, -1.0 close to body
front knee	-1.0 : 1.0	0.0 straight, 1.0 max bent, -1.0 max extended
back knee	-1.0 : 1.0	0.0 straight, 1.0 max bent, -1.0 max extended

To get the value of a joint, use `getJoint(`*jointName*`)`, where *jointName* is a string of the following form:

```
leg [front | back] [left | right] [rotator | elevator | knee]
head [tilt | pan | roll | nod]
tail [tilt | pan]
mouth
```

Examples:

```
robot.getJoint("leg front left rotator")
robot.getJoint("head roll")
robot.getJoint("mouth")
robot.getJoint("tail tilt")
```

To set the position of a joint use `setPose(`*jointName*`, `*amount*`)`, where *amount* is a floating-point value between -1.0 and 1.0 (see table above), and *jointName* is any of the possibilities from `getJoint()`.

Examples:

```
robot.setPose("leg front right elevator", -0.4)
robot.setPose("mouth", 0.35)
```

Alternatively, you can leave off the [elevator | rotator | knee] component of the leg joints and specify all three values at once. Similarly, you can specify both the pan and tilt of the tail with one command.

```
leg [front | back] [left | right], amountRotator, amountElevator, amountKnee
tail, amountPan, amountTilt
```

Examples:

```
robot.setPose("leg back left", 0.0, -0.2, 0.6)
robot.setPose("tail", 1.0, 0.5)
```

AIBO Sounds

You can make the AIBO speak like this:

```
robot.playSound("sound name")
```

Available sound names:

3barks, 3yips, barkhigh, barklow, barkmed barkreal, camera, catcry, catyowl, crash cutey, donkey, fart, glass, growl, growl2, grrr, howl, mew, ping, roar, skid, sniff, tick, toc, whiip, whimper whoop, yap, yipper

AIBO Sensors & Buttons

You can read the values of the AIBO's sensors and buttons in a similar way.

To get the value of a particular sensor use getSensor(*sensorName*) where *sensorName* is a string of the following form:

```
ir [near | far | chest]
accel [front-back | right-left | up-down]
power [remaining | thermo | capacity | voltage | current]
```

Examples:

```
robot.getSensor("ir near")
robot.getSensor("accel right-left")
```

This table explains the values returned by the sensors:

Sensor	**Unit**	**Range of values**
ir near	mm	50-500
ir far	mm	200-1500
ir chest	mm	100-900
accel (all)	positive-negative values	
power remaining	% of power remaining	0.0-1.0

power thermo	degrees Celsius
power capacity	milli-amp hours
power voltage	volts
power current	negative milli-amp hours

To read the raw value of a button use getButton(*buttonName*) where *buttonName* is a string of the following form:

paw [front | back] [left | right] : returns whether it's on or off
chin : pressure
head : pressure
body [front | middle | rear] : returns the amount of pressure being applied
wireless

Examples:

```
    robot.getButton("paw front left")
    robot.getButton("chin")
    robot.getButton("body middle")
```

Head Movement/Using the AiboCamera

To make use of the AIBO's head camera, type

```
robot.startDevice("AiboCamera")
```

at the Pyrobot command line. A window will pop up displaying the real-time image taken from the AIBO's camera. The AIBO Camera is fully integrated with Pyrobot's vision system, so you can filter, blobify, etc.

The robot's head is implemented as a standard pan/tilt/zoom device, but the zoom parameter is ignored. An additional parameter, roll, controls the angle of the head relative to the neck.

Examples:

```
  robot.ptz[0].pan(0.3)
  robot.ptz[0].roll(1.0)
```

The ranges of the pan/tilt/roll parameters are:

pan	-1.0 : 1.0	0.0 centered, 1.0 max left, -1.0 max right
tilt	-1.0 : 0.0	0.0 highest, -1.0 lowest
roll	0.0 : 1.0	0.0 lowest (straight ahead), 1.0 highest (about 45 degrees up)