INScore
Sensors Reference

D. Fober
GRAME
Centre national de création musicale

November 3, 2016

Contents

Chapter 1

Sensors

INScore supports various sensors, which can be viewed as objects or as signals. When created as a signal node, a sensor behaves like any signal but may provide some additional features (like calibration). When created as a score element, a sensor has no graphical appearance and provides specific sensor events and features.

Table **??** gives the list of supported sensors names.

name	values	description
accelerometer	x, y, z	acceleration on the x, y, and z axis
ambient light	light level	ambient light value
compass	azimuth,	azimuth is in degrees from magnetic north in a clockwise direction
gyroscope	x, y, z	the angular velocity around the x, y, and z axis
light	lux	the light level measured as lux
magnetometer	x, y, z	the raw magnetic flux density measured on th x, y and z axis
orientation	orientation	the device orientation
proximity	close	a boolean value
rotation	x, y, z	the rotation around the x, y and z axis
tilt	x, y	the amount of tilt on the x and y axis.

Table 1.1: Sensors names and description

NOTE

All the sensors won't likely be available on a given device. In case a sensor is not supported, an error message is generated at creation request and the creation process fails.

1.1 Sensors as signals

A sensor is viewed as a signal when created in a `signal` node using pre-defined signal names which are listed in table **??**. Values provided on different axis (e.g. acceleration on the x, y, and z axis) are available from the sensor subnodes, also listed in table **??**.

EXAMPLE

Creating a rotation sensor with a 200 values buffer size.

```
/ITL/scene/signal/rotation size 200;
```

Getting accelerometer values on the x axis.

```
/ITL/scene/signal/accelerometer/x get;
```

| sensor	signal name	subnodes
accelerometer	accelerometer	x, y, z
ambient light	ambientlight	*none*
compass	compass	*none*
gyroscope	gyroscope	x, y, z
light	light	*none*
magnetometer	magnetometer	x, y, z
orientation	orientation	*none*
proximity	proximity	*none*
rotation	rotation	x, y, [z]
tilt	tilt	x, y

Table 1.2: Sensor's signal names and subnodes

NOTE

The `rotation` sensor may or may not have a z component however, the *z* signal is always present but set to 0 when no z component is available. A specific message is provided to get the z component status (see section **??** p.**??**).

1.2 Sensors as nodes

A sensor is viewed as a regular INScore node when created outside a `signal` node and using one of the sensors types defined below. A sensor node has no graphical appearance but has the position attributes of an INScore object (x, y, z and scale).

sensorSet
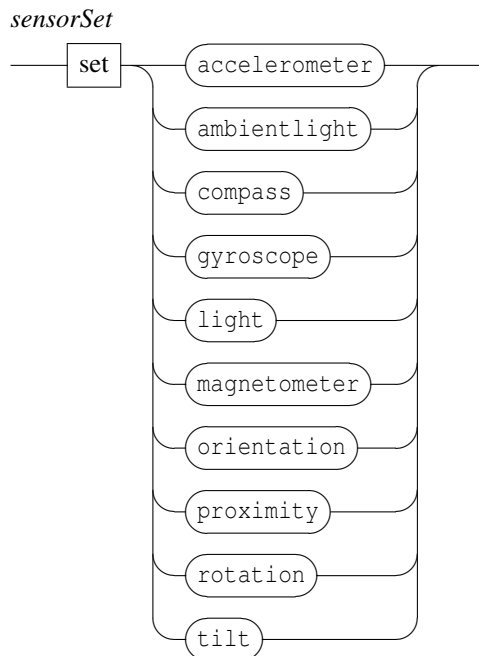
Values generated by a sensor are available using its x, y and z attributes. Depending on the sensor type, y and z may be useless. Note also that events generated in the context of a sensor have the variables $x, $y and $z set with the current sensor values (see section **??** p.**??**).

Creating a proximity sensor, querying it's value and watching the value changes.

```
 /ITL/scene/sensor set proximity;
/ITL/scene/sensor get x;
/ITL/scene/sensor watch newData (/ITL/scene/score show '$x');
```

1.3   Values

Values generated by the sensors depends on the sensor type and on the the sensor instance (i.e. whether created as signal or as node). Table **??** presents the values range for the node and the signal instances. The rationale is that nodes values are raw sensor values while signal values are mapped to the signal range i.e. [-1,1]. Actually, the mapping of the raw values depends on the sensor calibration that can be automatically or manually adjusted. See the section about calibration below.

sensor	node values	signal values	comment
accelerometer	[-v,v]	[-1,1]	depends on the calibration
ambient light	0,1,2,3,4,5	[-1,1]	see the note about ambient light below
compass	[-180,180]	[-1,1]	
gyroscope	[-v,v]	[-1,1]	depends on the calibration
light	[0,v]	[-1,1]	depends on the calibration
magnetometer	[-v,v]	[-1,1]	
orientation	0,1,2,3,4,5,6	[-1,1]	see the note about orientation below
proximity	0,1	[-1,1]	a boolean value mapped to -1, 1
rotation	x [-90, 90]	[-1,1]	
	y [-180, 180]	[-1,1]	
	z [-180, 180]	[-1,1]	
tilt	[-90,90]	[-1,1]	

Table 1.3: Sensor's values as node and as signal

NOTE ABOUT AMBIENT LIGHT

Ambient light is measured using discrete values ranging from 0 to 5, where 0 means undefined and 1 to 5 goes from dark to very bright.
A value v is mapped to $(v * 0.4 - 1)$

NOTE ABOUT ORIENTATION

Orientation is measured using discrete values ranging from 0 to 6, where 0 means undefined and 1 to 6 represents the following orientations:

- 1: the Top edge of the device is pointing up.
- 2: the Face of the device is pointing up.
- 3: the Left edge of the device is pointing up.
- 4: the Face of the device is pointing down.
- 5: the Right edge of the device is pointing up.
- 6: the Top edge of the device is pointing down.

A value v is mapped to $(v/3 - 1)$
In a given way and from values 2 to 5, the device may be viewed as rotating clockwise. A counter-clockwise option is also supported, see section **??** p.**??**.

1.4 Calibration

Calibration of sensor values may be viewed as scaling and makes use of the common object's `scale` attribute. By default, the scale value is 1.0 when the sensor is a regular node. For signal nodes, the default scale value is given by the table **??**. These values have been choosen to map the raw values to the signal range but of course this mapping depends on the device and may greatly vary. In order to accommodate these variations but also to cope with different requirements, scaling can be manually adjusted to any arbitrary value using the `scale` message, or automatically adjusted to measured peak values using the `autoscale` message.

| sensor	signal scale	comment
accelerometer	1/g	where g is the gravity on earth i.e. 9.81
ambient light	0.4	see the note about ambient light above
compass	1 / 180	
gyroscope	1 / 90	
light	1 / 200	an arbitrary lux value (considered as
magnetometer	10000	
orientation	1/3	see the note about orientation above
proximity	1.0	the `false` value is shifted to -1
rotation	1 / 180	for the x value, the scale is multiplied by 2
tilt	1 / 90	

Table 1.4: Sensor as signal default scaling

NOTE ABOUT AUTO-SCALING

Auto-scaling consists in measuring the peak of the absolute values of a sensor during a period. The sensor `scale` value is next adjusted to $1/peak$ (see also the sensor common messages in section **??**). Auto-scaling is supported by all the sensors, although

1.5 Sensor common messages

All sensors support a common set of message.

sensorCommon
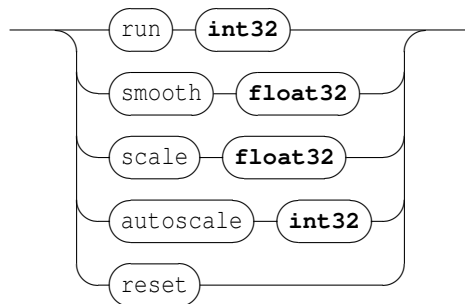
- `run`: takes a boolean value as parameter. When true, the sensor starts to generate values. Default value is false.
- `smooth`: applies exponential smoothing to the sensor values. At a time t, the sensor value is computed as: $s_t = \alpha.v_t + (1-\alpha).s_{t-1}$ where v_t is the current sensor value and $0 \leqslant \alpha \leqslant 1$. The parameter is the smoothing factor α. Default value is 1.

4

- `scale`: sensor values are multiplied by the scale. Default scale is dependent on the sensor type. See table **??** for the default scale values.
- `autoscale`: start or stop the auto-scaling process. Default value is false. See the note about auto scaling above. Note that a sensor must be running for the auto-scaling process to take effect.
- `reset`: reset the smoothing factor and the scale to their default values.

1.6 Sensor specific messages

1.6.1 Accelerometer sensor

accelerometerMsg
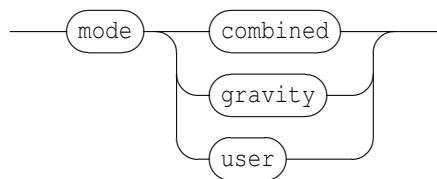
- `mode`: the acceleration mode controls how acceleration values are reported.
  - `gravity`: only the acceleration caused by gravity is reported. Movements of the device caused by the user have no effect other than changing the direction when the device is rotated.
  - `user`: only the acceleration caused by the user moving the device is reported, the effect of gravity is canceled out. A device at rest therefore should report values of, or close to, zero.
  - `combined`: both the acceleration caused by gravity and the acceleration caused by the user moving the device is reported combined.

  Default value is `combined`.

NOTE ABOUT MODES

Acceleration caused by gravity and acceleration caused by the user moving the device are physically impossible to distinguish because of general relativity. Most devices use sensor fusion to figure out which parts of the acceleration is caused by gravity, for example by using a rotation sensor to calculate the gravity direction and assuming a fixed magnitude for gravity. Therefore the result is only an approximation and may be inaccurate. The `combined` mode is the most accurate one, as it does not involve approximating the gravity.

### 1.6.2 Magnetometer sensor

magnetometerMsg
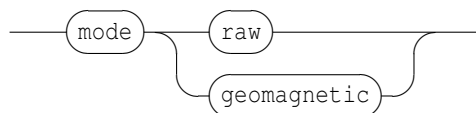
The magnetometer can report on either raw magnetic flux values or geomagnetic flux values. The primary difference between raw and geomagnetic values is that extra processing is done to eliminate local magnetic interference from the geomagnetic values so they represent only the effect of the Earth's magnetic field. This process is not perfect and the accuracy of each reading may change. Default value is `raw`.
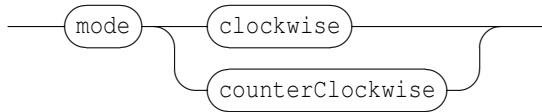
1.6.3 Rotation sensor

rotationMsg

```
──( get )──( hasZ )──
```
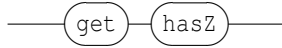
z angle availability of the rotation sensor can be queried using `hasZ`. The returned value is a boolean value.

1.6.4 Orientation sensor

orientationMsg

```
──( mode )──┬─( clockwise )────────┬──
            └─( counterClockwise )─┘
```

- `mode`: selects how the device position is mapped to successive values. Default value is `clockwise`. See table **??** for the detail of the positions and values.

value	clockwise	counter clockwise
1	Top edge up	Top edge up
2	Face up	Face up
3	**Left** edge up	**Right** edge up
4	Face down	Face down
5	**Right** edge up	**Left** edge up
6	Top edge down	Top edge down

Table 1.5: Device positions and values in different modes.

1.6.5 Tilt sensor

tiltMsg

```
──( calibrate )──
```
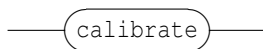
- `calibrate`: calibrates the tilt sensor: uses the current tilt angles as 0.