

RAGE

Tutorials

Input (Manual Polling)

Software & Documentation © The RAGE Team 2005

Version Relevance: JGE Release 0.2

CONTENTS

Are you reading the right tutorial?	3
Required Imports	3
Required Properties	3
Input Device Basics	4
Polling Device Components	5
Using the Default Keyboard	6
Using the Default Mouse	7

ARE YOU READING THE RIGHT TUTORIAL?

This tutorial is intended to show you how to use input system to read the states of input devices, so that it can be used to alter the state of your application.

Also covered is the specific use of keyboard and mouse devices and their enumerations.

The tutorial assumes that you have read the tutorial on the property manager.

REQUIRED IMPORTS

Input Package: `jge.hal.input`

REQUIRED PROPERTIES

<i>Property</i>	<i>Description</i>	<i>Example Value</i>
<code>class.InputPlatform</code>	Implementation of <code>jge.hal.input.InputPlatform</code> to use in JGE.	<code>plugin.hal.input.OSXInputPlatform</code>

INPUT DEVICE BASICS

Any device with buttons/keys, axes, or directionals can be considered an device in the input system. Devices can have any number of subcomponents which are any of the above three.

Subcomponents:

- **Key (Button)**
A component with two states: on and off. On is considered any state above 0.0f, off is 0.0f.
- **Axis**
A component with any number of states. The value of the state is always between -1.0f and 1.0f.
- **Directional**
A component with a limited number of states, values of the states are between -1.0f and 1.0f.

Before a device can be used it must be acquired. You can acquire all available devices by calling `acquireAllDevices()` on the `InputManager` or you can acquire just the devices you want.

Note: Calling `getDefaultKeyboard` and `getDefaultMouse` automatically acquire the keyboard/mouse if one is available.

POLLING DEVICE COMPONENTS

Polling a components state is simple. In the example below we will assume we are polling device with an ID 0, and we are polling its component that has an ID 2.

Updating the value in each component is handled by a system task and does not need to be handled independently. The value returned by the getState function will be updated once each cycle of the task loop (game loop).

```
// cache a reference to the input manager
InputManager im = InputManager.getInstance();
im.acquireAllDevices();

// get device '0'
InputDevice device = im.getDevice(0);

// get component '2' on device '0'
InputDeviceComponent component = device.getComponent(2);

// poll for the state
double polledState = component.getState();
```

USING THE DEFAULT KEYBOARD

Although one is not guaranteed to be available, input platforms are strongly recommended to provide a default keyboard and your code can be dependent on one existing.

The default keyboard can be accessed by calling the function below on the input manager.

```
jge.hal.input.InputManager
+getDefaultKeyboard():jge.hal.input.Keyboard

Keyboard k = InputManager.getInstance()
    .getDefaultKeyboard();
```

A Keyboard object has a map of enumerations to Key components. (See the `jge.hal.input.Keyboard` class in the JavaDocs for the enumerations.) Not all enumerations may be mapped, so it is important to check whether a key exists on the keyboard.

The example below shows how to get a Key component that is the escape key. The reference to the component can then be used in the previous chapter to poll.

```
// get the keyboard
Keyboard k = InputManager.getInstance()
    .getDefaultKeyboard();

// test if there is a keyboard
if(k == null)
{
    System.out.println("No Keyboard");
    System.exit(1);
}

// get the escape key
Key escape = k.getKey(Keyboard.KB_ESCAPE);

// print out if an escape key exists
if(escape == null)
{
    System.out.println("No escape Key");
}
else
{
    System.out.println("FOUND escape Key");
}
```

USING THE DEFAULT MOUSE

A mouse is never guaranteed, so you should provide failsafe in your code if one does not exist.

The default keyboard can be accessed by calling the function below on the input manager.

```
jge.hal.input.InputManager
+getDefaultMouse():jge.hal.input.Mouse

Mouse m = InputManager.getInstance().getDefaultMouse();
```

A Mouse object has a map of enumerations to Key and Axis components. (See the `jge.hal.input.Mouse` class in the JavaDocs for the enumerations.) Not all enumerations may be mapped, so it is important to check whether a key or axis exists on the keyboard. Typically with a mouse, if a mouse exists, it will have at least one button and at least two axes.

The example below shows how to get the x and y axes as well as button one of the mouse. The reference to the component can then be used in the previous chapter to poll.

```
// get the mouse
Mouse m = InputManager.getInstance().getDefaultMouse();

// test if there is a mouse
if(m == null)
{
    System.out.println("No Mouse");
    System.exit(1);
}

// get the axes and buttons
Axis xaxis = m.GetAxis(Mouse.MOUSE_AXIS_X);
Axis yaxis = m.GetAxis(Mouse.MOUSE_AXIS_Y);
Key button = m.getKey(Mouse.MOUSE_KEY_1);
```

