

RAGE

Tutorials

Input (Event Based)

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
Event Based Input Basics	4

ARE YOU READING THE RIGHT TUTORIAL?

This tutorial is intended to show you how to use input system to notify your application of state changes in the components of input devices.

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`

Event Input Package: `jge.hal.input.event`

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>

EVENT BASED INPUT BASICS

Using the event system reduces the amount of work you will need to do to find out about key presses or axis movement.

Any class you want to react to input changes needs to implement `jge.hal.input.event.InputObserver` and supply implementation for the functions provided. If a function is not needed, just provide an implementation with nothing in it (a null implementation). The class will then also need to be registered as an observer of the `jge.hal.input.event.InputObserverManager` class.

EXAMPLE CODE

```
public class MyClass implements InputObserver
{
    /** Escape Key on the Keyboard. */
    private Key escapeKey;

    public MyClass()
    {
        // cache references
        InputManager im = InputManager.getInstance();
        InputObserverManager iom =
            InputObserverManager.getInstance();

        // register this observer
        iom.addObserver(this);

        // get a reference to the keyboard
        Keyboard kb = im.getDefaultKeyboard();

        // get the escape key (assume its there)
        escapeKey = kb.getKey(Keyboard.KB_ESCAPE);
    }

    public void onKeyPress(Key key)
    {
        // if the escape key is pressed,
        // kill the TaskManager
        if(key == escapeKey)
        {
            TaskManager.getInstance().kill();
        }
    }

    // provide null functions
    public void onKeyDown(Key key) { }
    public void onKeyRelease(Key key) { }
    public void onAxisMove(Axis axis) { }
    public void onDirectionalChange(
        Directional directional) { }
}
```

