

Series 40 6th Edition extends the Nokia UI API with a new interface, the Force Sensitive Joystick API. Together with a force sensitive joystick, this new interface enables developing more sophisticated Java ME applications such as games and navigation applications with the MIDP low-level UI in Series 40. This article discusses these two components and what you need to know before you start developing joystick applications for Series 40.

Contents

- [1 Force-sensitive joystick in Series 40 6th Edition devices](#)
- [2 Force Sensitive Joystick API in Series 40 6th Edition](#)
- [3 Application development with Force Sensitive Joystick API](#)
- [4 Testing Java ME applications with Force Sensitive Joystick API](#)
- [5 Related links](#)

Force-sensitive joystick in Series 40 6th Edition devices

Certain Series 40 6th Edition devices, such as the Nokia 6260 Slide (see the below picture) have a force-sensitive joystick, an input device which detects 360 degree scrolling movement and force input generated by the end-user.

The force-sensitive joystick enhances user experience in two ways: First, it is a one-button control as instead of five separate navigation keys (Up, Down, Left, Right, and Middle) which have been used as controls for Java ME- based games and applications in earlier Series 40 Edition devices. Second, this single button detects 360 degree scrolling movement at force-dependent speed, enabling for example brake, acceleration and steering in racing games, and providing therefore a richer and more dimensional end-user experience.



Force Sensitive Joystick API in Series 40 6th Edition

Series 40 6th Edition devices which have a force-sensitive joystick can utilise the new Nokia UI API extension interface called Force Sensitive Joystick API. Java ME applications using this interface listen to joystick events generated by the end user. The usability of this interface can be checked by using `System.getProperty("com.nokia.mid.ui.joystick_event")` which returns "true" if the interface can be used.

The Force Sensitive Joystick API is an extension interface to the existing Nokia UI API (`com.nokia.mid.ui`). To use this interface, the following import definition needs to be done in the source code:

```
import com.nokia.mid.ui.JoystickEventListener;
```

The new joystick interface consists of one method:

```
void joystickEvent(int x, int y) {
    // use x and y coordinates here
}
```

In the source code, the `joystickEvent(int x, int y)` method is called when the joystick sends an event to the listener interface, `com.nokia.mid.ui.JoystickEventListener`. The joystick sends an event every time its position or the force applied to it changes. These events are represented by `x` (horizontal) and `y` (vertical) vector components. In the centre position, the value for an `(x, y)` event is `(0,0)`.

Since `x` and `y` coordinates are used as movement delta for determining both position and applied force, this sets certain limitations which should be noted especially from the user experience point of view. If, for example, the `(-5,5)` event is received, it means that the joystick is held in the left and up diagonal. Now, if the `(-25,25)` event is sent, it means that the joystick is still held in the same direction of `(-5,5)`, regardless of more force applied for the new event.

The maximum and minimum values of `x` and `y` coordinates are `-127` and `127`.

Application development with Force Sensitive Joystick API

The Force Sensitive Joystick API can only be used with MIDP's low-level user interface: If the MIDlet needs to receive joystick events, it must provide an implementation of the event listener interface from a Canvas class or subclass. The Canvas will continue to receive standard key / game key events. In addition, the joystick events are sent only if the Canvas is in visible state.

The following code snippet illustrates the basic usage of the Force Sensitive Joystick API.

```
/* -----
 * JoystickTest.java
 *
 * Midlet for processing joystick commands
 *
 *----- */

import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Graphics;
import com.nokia.mid.ui.FullCanvas;
import com.nokia.mid.ui.JoystickEventListener;

public class JoystickTest extends MIDlet
{
    private Display display; // For current display of midlet
    private JoystickCanvas canvas; // Full Canvas Class with functionality
    private Form myform; // Form to show initial results

    public JoystickTest()
    {
        display = Display.getDisplay(this);
        myform = new Form("Test Form");
        canvas = new JoystickCanvas();
    }
    /*
     * At startup, this midlet will test whether the necessary
     * functionality is supported on this device or not.
     * @see javax.microedition.midlet.MIDlet#startApp()
     */

    protected void startApp()
```

```

{
    String val = System.getProperty("com.nokia.mid.ui.joystick_event");
    if(val != null){
        setCurrentDisplayCanvas );
    }
    else{
        append("Joystick event not supported on this device");
        setCurrentDisplayForm );
    }
}
}
protected void pauseApp(){ }

protected void destroyApp( boolean unconditional )
{
    notifyDestroyed
}

} // end of class JoystickTest

/*-----
* Class JoystickCanvas
*
* This class actually implements the JoystickEventListener interface
* and its functionality as a whole, using the FullCanvas class.
*-----*/
class JoystickCanvas extends FullCanvas implements JoystickEventListener
{
    private int height, y_pos;    // height value along y-axis
    private int width, x_pos;    // width value along x-axis
    private String coordinates; // x,y coordinates are converted to string to display on canvas.
    private int H_MID, W_MID;    // Midpoint of total height and total width.

    /*-----
    * Constructor
    *-----*/
    public JoystickCanvas()
    {
        height = y_pos = this.getHeight(); // Full height for this device (Nokia_6600 = 480 )
        width = x_pos = this.getWidth(); // Full width for this device (Nokia_6600 = 320 )
        H_MID = height / 2; // Half of this height 240
        W_MID = width / 2; // Half of this width 160
    }

    /*-----
    * Paint the text representing the key code
    *-----*/
    protected void paint(Graphics graphics)
    {
        // Clear the background (to light grey)
        graphics.setColor(200, 200, 200);
        graphics.fillRect(0, 0, width, height);

        // Drawing Axis lines on Canvas
        drawAxes(graphics);

        // Paint a red coloured box, each time with new x,y coordinates.
        graphics.setColor(255,0,0);
        graphics.fillRect((x_pos+ W_MID) , (-y_pos+H_MID), 10, 10);
        graphics.drawRect((x_pos+ W_MID) , (-y_pos+H_MID), 10, 10);

        // String representation of x,y coordinates is painted on canvas's left_bottom.

```

```

        coordinates = "["+ x_pos +"," + y_pos +"]"; //320x480
        graphics.drawString( coordinates , 0 , height , Graphics.BOTTOM | Graphics.LEFT);
    }

    /* -----
    * This method will draw the horizontal and vertical axis
    * in the middle of the screen by using height, width values.
    * ----- */
    public void drawAxes(Graphics g){

        g.setStrokeStyle(Graphics.DOTTED);
        g.setColor(0,0,255); // blue coloured axes
        g.drawLine(0, height/2, width - 1, height/2); // drawing horizontal line
        g.drawLine(width/2, 0, width/2, height - 1); // drawing vertical line
    }

    /*-----
    * Joystick Event handler method.
    *
    * It fetches the x,y coordinates of current position
    * and paints it on canvas using x_pos , y_pos.
    *-----*/
    public void joystickEvent(int x, int y)
    {
        this.x_pos = x;
        this.y_pos = y;
        repaint();
    }

} // End of JoystickCanvas

```

Testing Java ME applications with Force Sensitive Joystick API

In addition to the actual Nokia 6260 Slide device, Java applications with Force Sensitive Joystick API can be tested by using the Series 40 6th Edition SDK emulator.

Any USB-based joystick should suit for testing the API with the SDK emulator, preferably with a Windows XP supported plug-and-play convenience to make it work without any additional steps.

If plug-and-play is not available, vendor drivers needs to be installed as prerequisite to make the joystick work with the SDK emulator.

Related links

- Javadocs on Force Sensitive Joystick API (com.nokia.mid.ui.JoystickEventListener) in Forum Nokia Java Developer's Library:
http://library.forum.nokia.com/index.jsp?topic=/Java_Developers_Library/GUID-237420DE-CCBE-4A74-A1

- Series 40 6th Edition SDK download:
http://www.forum.nokia.com/info/sw.nokia.com/id/cc48f9a1-f5cf-447b-bdba-c4d41b3d05ce/Series_40_Platform_SDK_download
- Nokia 6260 Slide device details: http://www.forum.nokia.com/devices/6260_slide
- How to drive radio-controlled car with Nokia N97 or Nokia 6260 Slide devices (Force Sensitive Joystick API use case):
http://wiki.forum.nokia.com/index.php/How_to_drive_radio-controlled_car_with_Nokia_N97_or_Nokia_6260_Slide_devices