NOKIA Developer

# Series40 设备上多点触摸交互的处理

## Introduction

中文

在 Java Runtime 2.0.0 for Series 40 版本中添加了对多点触摸API的支持，那么本文的目的就是介绍如何使用多点触摸API，处理交互的事件。

多点触摸API允许在基于Canvas的MIDlet中处理多点触摸事件，比如说GameCanvas， FullCanvas等，通过API我们可以注册以便能够接收到多点触摸事件。

## API介绍

每一个触摸点是包括3个重要的数值：

- 一个唯一的ID
- 当前的坐标值，包括X， Y
- 当前触点的状态

其中触摸的状态有3种：

- POINTER_PRESSED
- POINTER_DRAGGED
- POINTER_RELEASED

多点触摸的API很简单，主要包括了下面的：

-MultipointTouch 该类主要用来访问多个触电的相关坐标，状态等信息，设备所支持的触点数等信息，绑定或移除多点触摸监听者等。

-MultipointTouchListener 接收多点触摸事件，并做出相应处理。

## API的使用

- 判断当前设备是否支持 MultiTouch API

Nokia SDK中提供了属性com.nokia.mid.ui.multipointtouch.version 来获取API版本信息。

```
if (System.getProperty("com.nokia.mid.ui.multipointtouch.version") != null) {
    // API is supported: Can implement multipoint touch functionality
} else {
    // API is not supported: Cannot implement multipoint touch functionality
}
```

如何获取特定设备所支持的最大触点数呢： 可以使用- MultipointTouch.getMaxPointers

- 获取MultipointTouch实例

```
MultipointTouch mpt = MultipointTouch.getInstance();
```

- 为MIDlet注册MultipointTouchListener

```
public class MainCanvas extends Canvas implements MultipointTouchListener
{
```


(C) Copyright Nokia 2014. All rights reserved.

```
    public MainCanvas() {
        // ...
        mpt.addMultipointTouchListener(this);
    }
......
}
```

- 处理多点触摸事件

> 从函数pointersChanged(int[] pointerIds)可以看出参数仅仅是一个触摸点ID的数组，然后我们是通过ID值使用MultipointTouch来获取触点的相关信息。

这里需要注意的，参数pointerIds 这个数组仅仅是包含了状态，位置有变化的触摸点的ID，没有变化的并不会被包含在该数组中。

```
public void pointersChanged(int[] pointerIds) {
    for(int i=0; i<pointerIds.length; i++) {        // Loop through the changed touch
points
    {
        int pointerId = pointerIds[i];              // Get the touch point ID
        int state = MultipointTouch.getState(pointerId);  // Get the touch point state
        // Get the touch point x and y coordinates
        int x = MultipointTouch.getX(pointerId);
        int y = MultipointTouch.getY(pointerId);

        // Handle the UI update based on the touch point state, ID and coordinates
        switch(state) {
            case MultipointTouch.POINTER_PRESSED:   // A new finger was pressed against
the screen
                drawTouch(pointerId, x, y); break;
            case MultipointTouch.POINTER_DRAGGED:   // A pressed finger was dragged over
the screen
                drawTouch(pointerId, x, y); break;
            case MultipointTouch.POINTER_RELEASED:  // A pressed finger was lifted from
the screen
                break;
                 }
             }
        }
```

## 实例演示

一个是使用多点触摸功能的小游戏，详情见：


(C) Copyright Nokia 2014. All rights reserved.

多点触摸实例演示

# 相关链接

- Portal:Java ME (Chinese)


(C) Copyright Nokia 2014. All rights reserved.