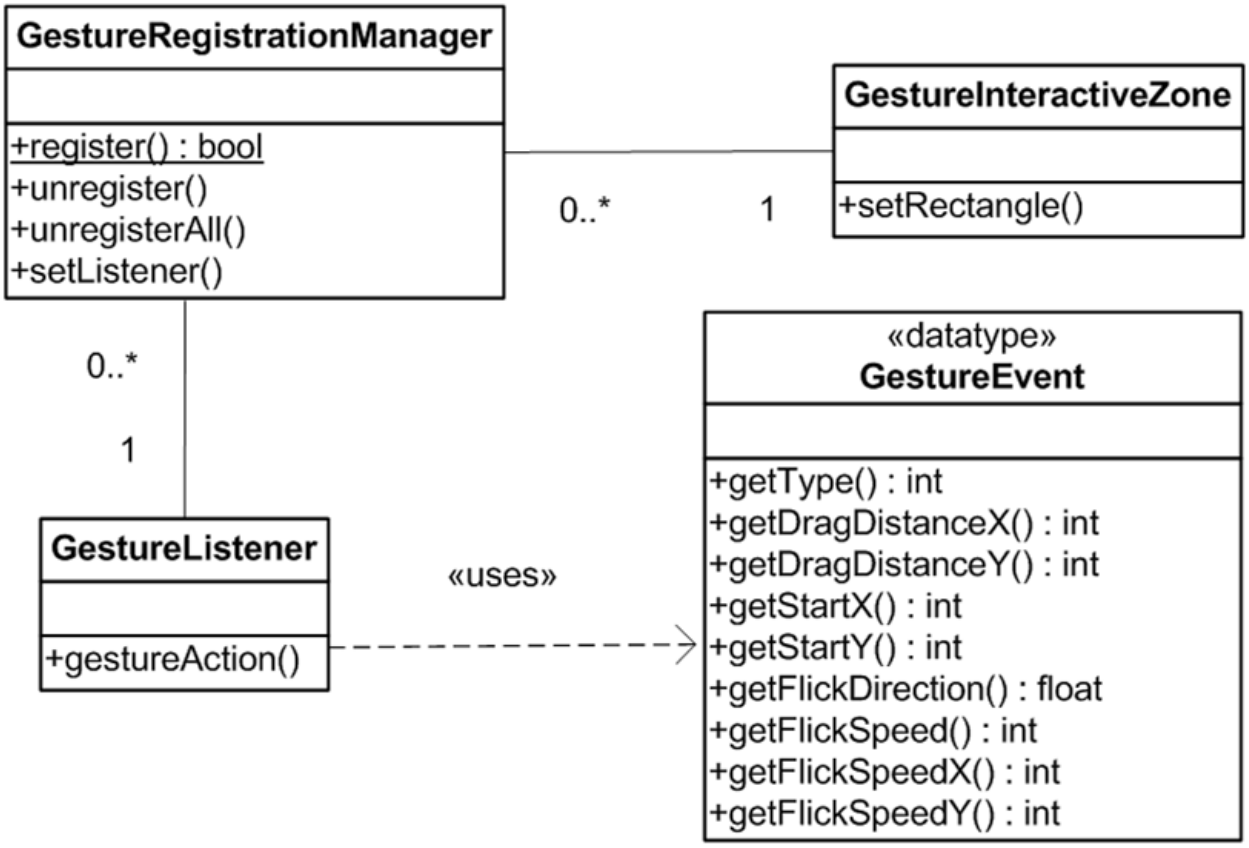
# Series 40 Touch API - Gesture API (Chinese)

利用触摸屏的手势操会使得应用程序的用户体验大大提升，比如用Drag进行屏，用Flick进行滚动等。

从Series 40_6th_Edition_FP1开始，诺基亚Series 40 SDK中加入了com.nokia.mid.ui.gestures 包，对触摸屏手势进行支持。开发者无需再自己实现手势识别引擎。Nokia Gesture API包由两个接口和两个类组成。




接口

| | |
|---|---|
| GestureEvent | The GestureEvent interface class is used by an application to receive gesture recognition events from the platform. |
| GestureListener | This interface is used by applications which need to receive gesture events from the implementation. |

类

| | |
|---|---|
| GestureInteractiveZone | The GestureInteractiveZone class is used by an application to define an area of the screen that reacts to a set of specified gestures. |
| GestureRegistrationManager | The GestureRegistrationManager class provides the ability to register a GestureListener to be notified when a gesture event occurs within a container. |

Gesture API采用了观察者设计模式。

## 第一步：创建GestureInteractiveZone实例

GestureInteractiveZone定义了一个可以接收手势事件的区域。缺省的GestureInteractiveZone包括整个屏幕。GestureInteractiveZone同时注册了MIDlet响应哪些手势事件。

// 创建一个相应所有手势事件的GestureInteractiveZone 对象 GestureInteractiveZone giz = new GestureInteractiveZone( GestureInteractiveZone.GESTURE_ALL ); // 设置相应区域 giz.setRectangle( x, y, width, height);


(C) Copyright Nokia 2014. All rights reserved.

在GestureInteractiveZone中定义了可被识别的手势：

```
static int GESTURE_ALL Constant for All Gesture Events.

static int GESTURE_DRAG Constant for the Drag Gesture.

static int GESTURE_DROP Constant for the Drop Gesture.

static int GESTURE_FLICK Constant for the Flick Gesture.

static int GESTURE_LONG_PRESS Constant for the Long Press Gesture.

static int GESTURE_LONG_PRESS_REPEATED Constant for the Long Press Repeated Gesture.

static int GESTURE_TAP Constant for the Tap Gesture.
```

## 第二步，实现GestureListener接口

GestureListener接口只定义了一个方法：gestureAction()， 当系统的手势识别引擎探测到在第一步中注册了的手势后，这个方法会被调用。该方法每次被调用时会接收到一个GestureEvent实例。GestureEvent中保存有最后接收到的手势事件的参数。

```
public void gestureAction(Object container, GestureInteractiveZone gestureZone,
GestureEvent gestureEvent)
{
     switch( event.getType() ) {
            case GestureInteractiveZone.GESTURE_TAP:
                 ...;
                 break;
            case GestureInteractiveZone.GESTURE_LONG_PRESS:
            case GestureInteractiveZone.GESTURE_LONG_PRESS_REPEATED:
            case GestureInteractiveZone.GESTURE_DRAG:
            case GestureInteractiveZone.GESTURE_DROP:
            case GestureInteractiveZone.GESTURE_FLICK:
        }
}
```

GestureEvent接口中定义了大量的get方法。 对于所有的手势事件都可以得到x，y坐标；

```
int getFlickSpeedX() Query for the Flick gesture events speed in horizontal direction.

int getFlickSpeedY()  Query for the Flick gesture events speed in vertical direction.
```

对于DRAG 和 DROP事件还可以分别得到x和y方向上变化的距离；

```
int getDragDistanceX()
          Query for the Drag & Drop gesture events movement in horizontal direction
since last drag gesture.
int getDragDistanceY()
          Query for the Drag & Drop gesture events movement in vertical direction since
last drag gesture.
```

对于FLICK事件，可以得到移动的速度和方向；


(C) Copyright Nokia 2014. All rights reserved.

```
 float getFlickDirection()      Query for the Flick gesture events direction.

 int getFlickSpeed()    Query for the Flick gesture events speed in actual flick
direction.

 int getFlickSpeedX()   Query for the Flick gesture events speed in horizontal
direction.

 int getFlickSpeedY()   Query for the Flick gesture events speed in vertical direction.
```

### 第三步，注册GestureInteractiveZone和Listener

GestureRegistrationManager类

```
static boolean register(java.lang.Object container, GestureInteractiveZone
gestureInteractiveZone)
          Register a gesture interactive zone to a container.
static void setListener(java.lang.Object container, GestureListener listener)
          Add a listener to the a container.
```

这两个方法的参数中都包括了一个容器类（Canvas 或者 CustomItem）。下面的代码演示了如何注册GestureInteractiveZone和Listener：

```
// 注册GestureInteractiveZone
Canvas canvas = new GestureCanvas();
GestureRegistrationManager.register( canvas, giz );

//注册 Listener
GestureRegistrationManager.setListener(canvas, this);
```

使用Gesture API时应注意：

- 1． 不要在gestureAction(...)方法中阻塞UI线程。
- 2． 不要将gestureAction(...)以外的变量指向GestureEvent实例。

## 代码示例

Media:TouchSample.zip

## 相关链接

- Java ME 中文 Homepage


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.