Gesture

Programming Guide

Version 2.0

Revision History

Version	Date	Description
1.0	July 24, 2013	Created this document.
2.0	January 08, 2014	Added speed.

Table of Contents

1. Overview

Gesture allows you to use events generated by hand movements in front of Samsung smart devices in your application.

You can use Gesture tosupplement device motion events.

1.1. Architecture

The following figure shows the Gesture architecture.

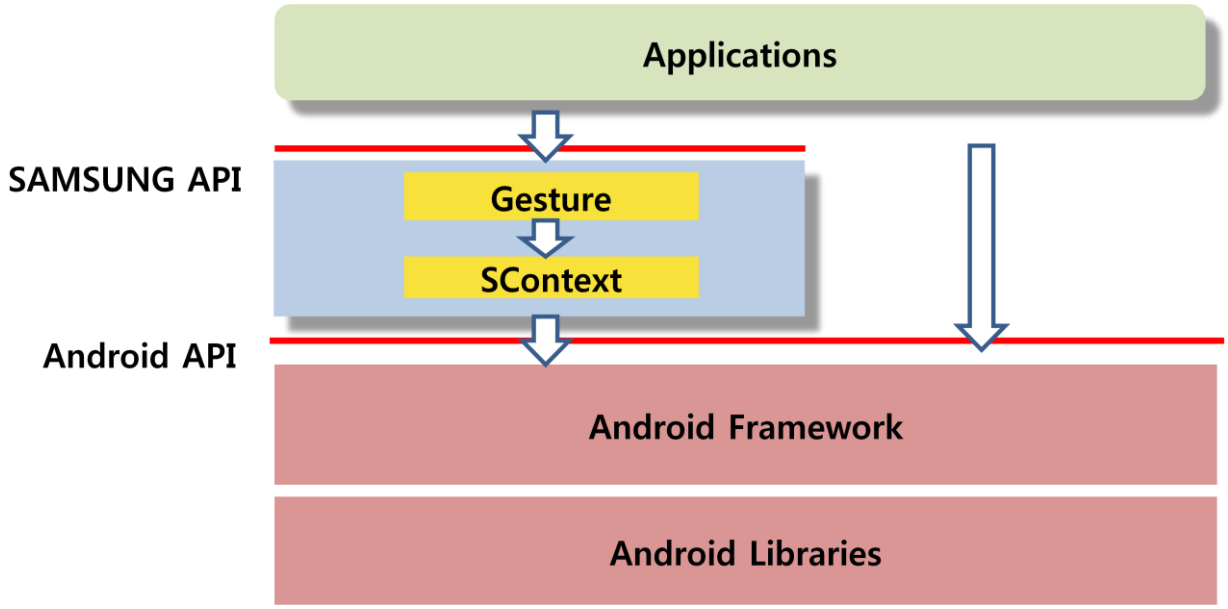

Figure 1: Gesture architecture

The architecture consists of:

- **Applications:** One or more applications that use Gesture.
- **Gesture:**Components for recognizing hand gestures.

1.2. Class Diagram

The following figure shows the Gesture classes and interfaces that you can use in your application.

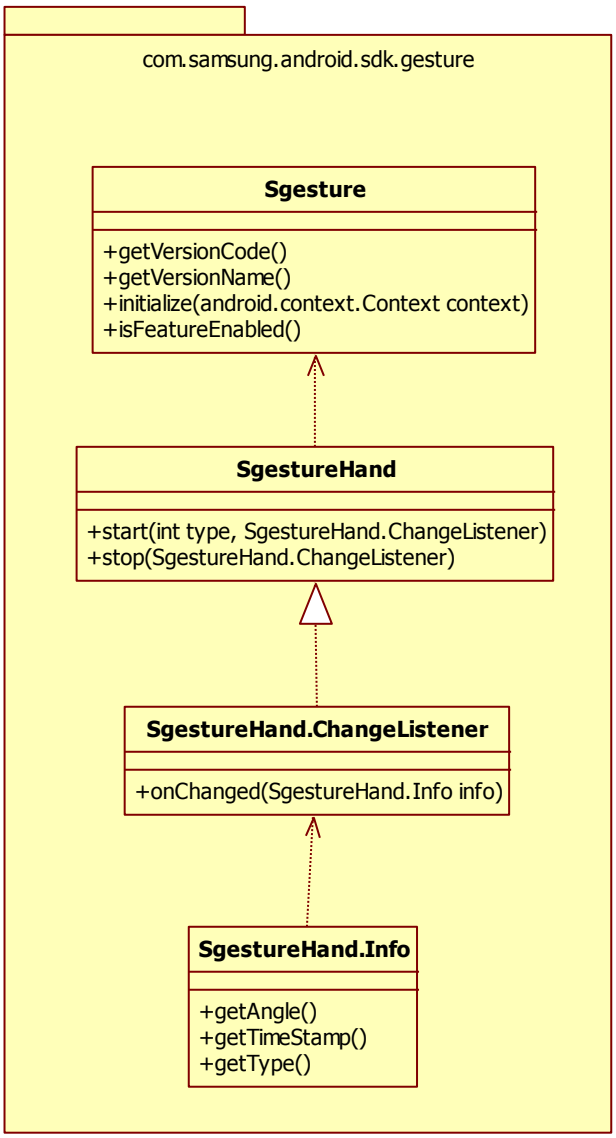

Figure 2: Gesture classes and interfaces

The Gesture classes and interfaces include:

- **Sgesture:**Initializes the Gesture package.
- **SgestureHand:**Recognizes hand movements using device sensors.
- **SgestureHand.Info:**Contains hand gesture eventinformation.
- **ChangeListener:**Listens for hand gesture events.

1.3. Supported Platforms

- Android 4.3 (Android API level 18)

1.4. Supported Features

Gesture supports the following features:

- Recognizing hand movements using device sensors

1.5. Components

- Components:
  - gesture-v2.0.0.jar
- Imported packages:
  - com.samsung.android.sdk.gesture

1.6. Installing the Package for Eclipse

To install Gesture for Eclipse:

1. Add the gesture-v2.0.0.jar file to the libs folder in Eclipse.

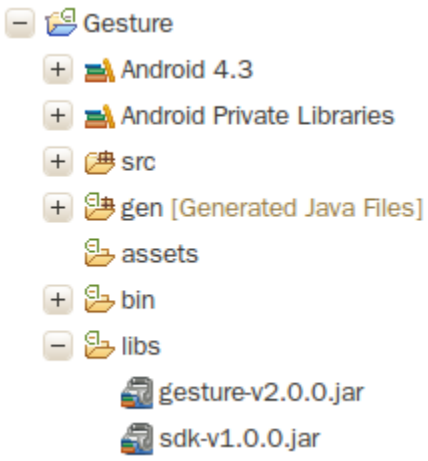

Figure 3: libs folder in Eclipse

2. Hello Gesture

Hello Gesture is a simple program that:

1. RegistersSgesture and SgestureHand instances.
2. Implementsand registers a ChangeListener instance.
3. Implements the `ChangeListener.onChanged()`method for receiving hand gesture events.
4. Stopsthe ChangeListener instance.

```
publicclassMainActivityextends Activity {
	privatestaticfinal String TAG = "HelloGesture";
	privateSgestureHandmGestureHand;
	privateSgesturegesture;

	@Override
	protectedvoidonCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		gesture = newSgesture();
		try {
	gesture.initialize(this);
		  } catch (IllegalArgumentException e) {
		     //Error handling
		  } catch (SsdkUnsupportedException e) {
		     //Error handling
		  }
		 if(gesture.isFeatureEnabled(Sgesture.TYPE_HAND_PRIMITIVE)){
		       //Feature check
		}
		mGestureHand = newSgestureHand(Looper.getMainLooper(),gesture);
		 //Start hand gesture
		 mGestureHand.start(Sgesture.TYPE_HAND_PRIMITIVE, changeListener);
	  }


	@Override
	protectedvoidonDestroy() {
	   //Stop hand gesture
	mGestureHand.stop(changeListener);
	   super.onDestroy();
	}

		privatefinalSgestureHand.ChangeListenerchangeListener =
			newSgestureHand.ChangeListener() {

		@Override
		publicvoidonChanged(Info info) {
				// TODO Auto-generated method stub
			if(info.getType() == Sgesture.TYPE_HAND_PRIMITIVE){
				System.out.println("Hello Gesture!!");
			}
		}
	};
}
```

3. Using the Sgesture Class

The Sgestureclass provides the following methods:

- `initialize()` initializes Gesture.You need to initialize the Gesture package before you can use it. If the device does not support Gesture, SsdUnsupportedException is thrown.
- `getVersionCode()` gets the Gestureversion number as an integer.
- `getVersionName()` gets the Gestureversion name as a string.
- `isFeatureEnabled(int type)` checks if a Gesturepackage feature is available on the device.

```
Sgesture gesture = newSgesture();

try {
       gesture.initialize(this);
} catch (IllegalArgumentExceptione) {
//Error handling
} catch (SsdkUnsupportedException e) {
//Error handling
}
```

3.1. Using the initialize() Method

The `Sgesture.initialize()` method:

- Initializes the Gesture package
- Checks if the device is a Samsung device
- Checks if the device supports the Gesture package
- Checks if the Gesture package libraries are installed on the device

```
void initialize(Context context) throwsSsdkUnsupportedException
```

If the Gesture package fails to initialize, the `initialize()` method throws an SsdkUnsupportedException exception. To find out the reason for the exception, check the exception message.

3.2. Handling SsdkUnsupportedException

If an SsdkUnsupportedException exception is thrown, check the exception message type using `SsdkUnsupportedException.getType()`.

The following types of exception messages are defined in the Sgesture class:

- **VENDOR_NOT_SUPPORTED:**The device is not a Samsung device.
- **DEVICE_NOT_SUPPORTED:**The device does not support the Gesture package.

3.3. Checking the Availability of Gesture Package Features

You can check if a Gesture package feature is supported on the device with the `isFeatureEnabled()` method. The feature types are defined in the Sgesture class.Pass the feature type as a parameter when calling the `isFeatureEnabled()` method. The method returns a Boolean value that indicates the support for the feature on the device.

```
booleanisFeatureEnabled(int type);
```

4. Using the Gesture Package

This section describes how to use the Gesture package in your application.

4.1. Receiving Data from the Gesture Package

To initialize the Gesture package and receive hand movement data:

1. Create anSgesture instance.
2. Pass the Sgesture instance as a parameter to create anSgestureHandinstance.
3. Call `start()` to register aChangeListener instance for the specified hand gesture type. When Gesture starts, SgestureHand sends a callback to the ChangeListener.
4. In the onChanged(`Info info`)method, handle thehand gesture events.
5. Call `stop()`to remove the ChangeListener instance.

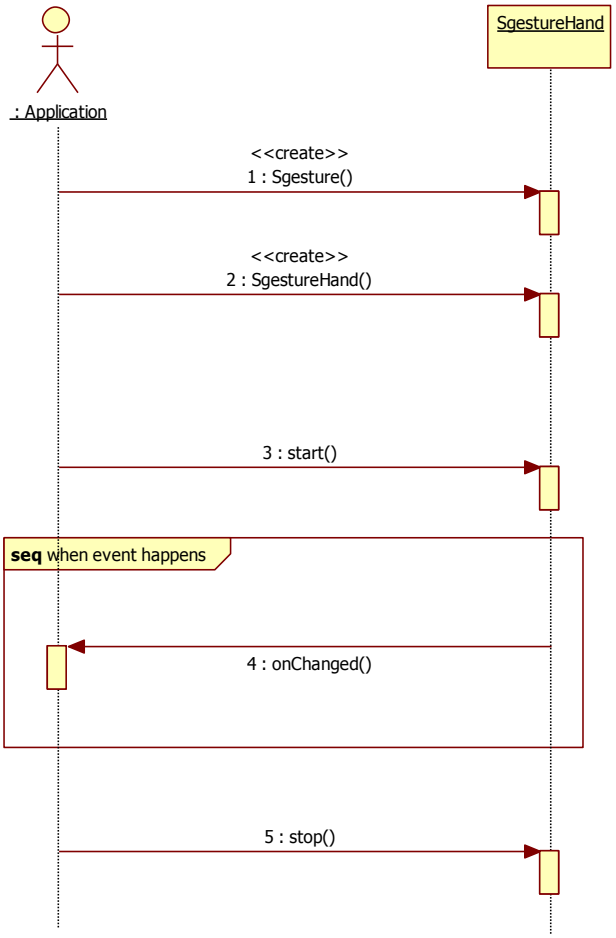

Figure 4: Receiving data from Gesture

```
SgestureHandmGestureHand;
Sgesturegesture;

// Initialize
      Sgesture gesture = newSgesture();
      try {
          gesture.initialize(this);
      } catch (IllegalArgumentExceptione) {
          //Error handling
      } catch (SsdkUnsupportedException e) {
          //Error handling
      }
      // Create SgestureHand instance
      mGestureHand = newSgestureHand(Looper.getMainLooper(),gesture);

// Implement SgestureHand.ChangeListener
       privatefinalSgestureHand.ChangeListenerchangeListener =
          newSgestureHand.ChangeListener() {

       @Override
       publicvoidonChanged(Info info) {
              // TODO Auto-generated method stub
              if(info.getType() == Sgesture.TYPE_HAND_PRIMITIVE){
              System.out.println("Hello Gesture!!");
              }
       }
};

//Add SgestureListenerfor the specified SgestureHandtype
mGestureHand.start(Sgesture.TYPE_HAND_PRIMITIVE, changeListener);

//Remove the registered SgestureListener
mGestureHand.stop(changeListener);
```

4.2. Using the Hand Gestures

The HAND_PRIMITIVE gesture type helps recognize hand gestures and returngesture data.When the user movesthe hand above the gesture sensor, you can get the hand gesture details:

- Use the `SgestureHand.Info.getAngle()`method to get the angle of the handat the end point of thegesture.

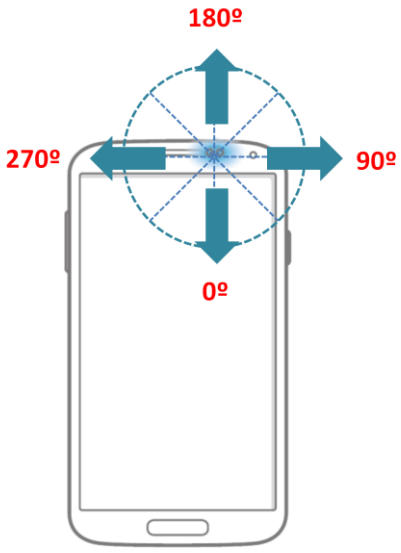

Figure 5: Position of the gesture Sensor and the gesture angle direction

The gesture sensor is placed atthe right-top side of the device. The gesture angle is measured in radians. The value of the angle is zero near the device displayand increasesin the counter-clockwise direction.

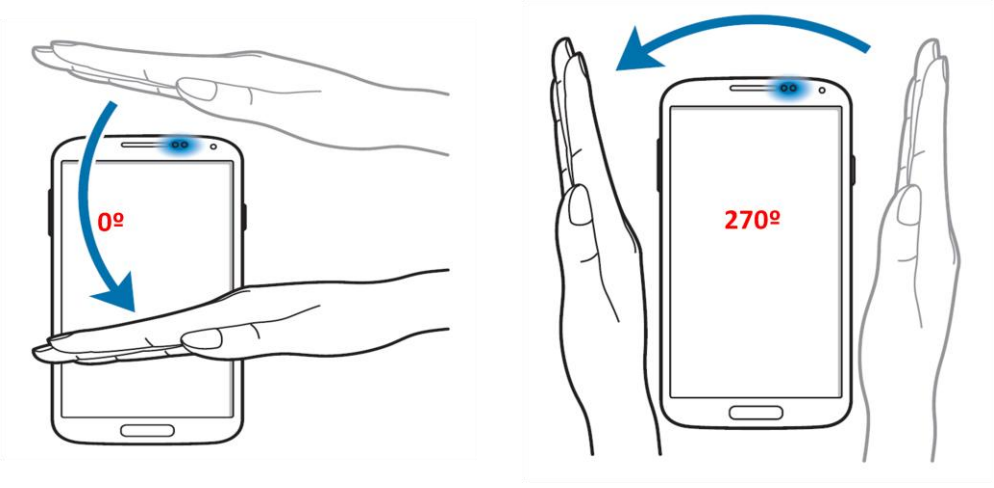

Figure 6: Gesture angles

- Use the `SgestureHand.Info.getSpeed()` method to get the speed of the hand movement. Gesture measures the speed on a scale of 0 to 100, where 0 is very slow and 100 is very fast. If `getSpeed()` returns -1, the device doesnot support speed measurements.

- Use the SgestureHand.Info.getTimeStamp() method to get the timestamp of the hand gesture. You can measure a duration by comparing the timestamp against another timestamp from the same process on the same device. The timestamp does not have a defined correspondence to wall clock times. The zero value is typically whenever the device was last booted. You can use System.currentTimeMillis() to get the current time.

```
publicclassMainActivityextends Activity {
       privatestaticfinal String TAG = "HelloGesture";
       privateSgestureHandmGestureHand;
       privateSgesturemGesture;

       @Override
       protectedvoidonCreate(Bundle savedInstanceState) {
                …
mGestureHand = new SgestureHand(Looper.getMainLooper(),mGesture);
mGestureHand.start(Sgesture.TYPE_HAND_PRIMITIVE,changeListener);
        }


       @Override
       protectedvoidonDestroy() {
              …
        }

         privatefinalSgestureHand.ChangeListenerchangeListener =
                              newSgestureHand.ChangeListener() {

              @Override
              publicvoidonChanged(Info info) {
                           // TODO Auto-generated method stub
                           if(info.getType() == Sgesture.TYPE_HAND_PRIMITIVE){
                           int angle = info.getAngle();
                           intspeed = info.getSpeed();
                           }
              }
       };
}
```

Copyright

Though every care has been taken to ensure the accuracy of this document, Samsung Electronics Co., Ltd.cannot accept responsibility for any errors or omissions or for any loss occurred to any person,whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

Samsung Electronics Co. Ltd.may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document.The furnishing of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of Samsung Electronics Co. Ltd.

The document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

For more information, please visit http://developer.samsung.com/