三星 S Pen Remote SDK 适配指导

上一篇我们介绍了 S Pen 悬空操作的 Air Actions 空中动作适配方法，通过 S Pen Framework 可以轻松识别向上，向下，向左，向右，顺时针和逆时针响应方向动作。如果应用需要获得 S Pen 运动的原始数据 ，设计自定义的手势轨迹，或者利用相对位置实现更多的功能，就需要使用 S Pen Remote SDK。本文将介绍 S Pen Remote SDK 使用方法。S Pen Remote SDK 可以帮助应用获取到 S Pen 按键和运动轨迹变化的事件，应用可利用这些信息实现自己独有的或者更强大的功能。

1. S Pen Remote SDK 概述

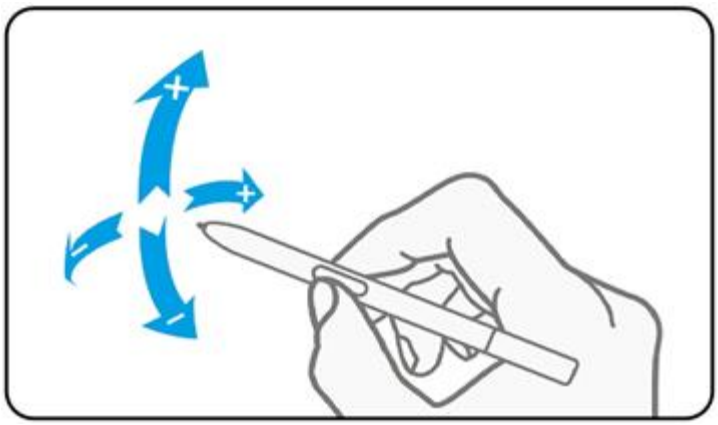
1.1 支持的平台

目前仅三星 Galaxy Note 系列支持 S Pen Remote SDK。S Pen Remote SDK 依赖于内部 Android framework 的静态 Java 库，所以此程序仅在支持这些模块的设备上运行。

1.2 支持的功能

S Pen Remote SDK 为应用提供了识别运动坐标和识别点击按钮的功能。

- **检查是否按下或释放了侧面按钮。**
- **识别运动坐标**
  - 设备二维运动
  - 当前位置与之前位置的相对值
  - x 坐标的正值表示向右移动
  - y 坐标的正值表示向上移动
  - 值范围：-1.0~1.0



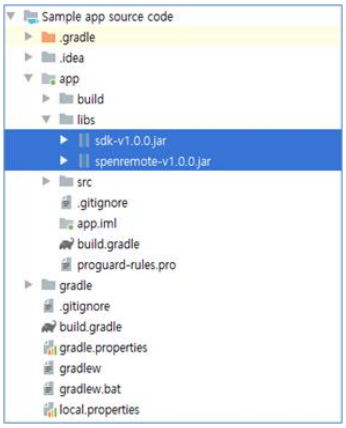
1.3 组件

- **组件**
  - spenremote-v1.0.0.jar
  - sdk-v1.0.0.jar

- **导入 package：**
  - com.samsung.android.sdk.penremote

1.4 Android Studio 中如何安装 Remote SDK

- 将 spenremote-v1.0.0.jar 和 sdk-v1.0.0.jar 文件添加到 Android Studio 中的 libs 文件夹中。



- 在 build.gradle 中添加 SDK 库文件夹依赖项

```
dependencies{
    compile fileTree(include: '*.jar', dir: 'libs')
    ...
}
```

2. SpenRemote 使用方法

SpenRemote 是一个类，提供以下方法：

- getInstance()获取 SpenRemote 的对象。
- getVersionCode()获取 S Pen Remote 版本号，为整数。
- getVersionName()获取 S Pen Remote 版本名称，为字符串。
- isFeatureEnabled（int type）检查设备是否有 S Pen Remote 功能。
- connect()用于连接 S Pen Framework。需要连接到 S Pen Framework 才能使用 S Pen Remote Features。
- disconnect()用于断开 S Pen Framework 的连接。

2.1 检查 SpenRemote 功能的可用性

可以使用 isFeatureEnabled()方法检查设备是否支持 SpenRemote 功能。功能类型在 SpenRemote 类中定义。调用 isFeatureEnabled()方法时，功能类型作为参数传递。该方法返回一个布尔值来表示是否支持该功能。

```
public boolean isFeatureEnabled(int type)
```

以下类型在 SpenRemote 类中定义为常量：

- **FEATURE_TYPE_BUTTON**
- **FEATURE_TYPE_AIR_MOTION**

2.2 连接到 S Pen Framework

需要连接到 S Pen Framework 来使用 SpenRemote 中支持的功能。要连接到 S Pen Framework，需要创建 **ConnectResultCallback** 并调用 **connect()**方法。如果已建立连接，则 connect()方法将不响应。因此，在尝试连接之前，使用 **isConnected()**方法检查连接状态。

```
SpenUnitManager mSpenUnitManager = null;
...
SpenRemote spenRemote = SpenRemote.getInstance();
if (!spenRemote.isConnected()) {
    spenRemote.connect(getContext(),
        new SpenRemote.ConnectionResultCallback() {
        @Override
        public void onSuccess(SpenUnitManager manager) {
            mSpenUnitManager = manager;
            ...
        }
```

```
            @Override
            public void onFailure(int error) {

            }
        });
    }
```

如果连接成功，则调用 ConnectResultCallback.onSucces()，并传递 SpenUnitManager 实例。SpenUnitManager 提供了设置事件监听器的方法，来监视 S Pen 中嵌入式单元的事件。

如果连接失败，则会将错误代码传递给 ConnectResultCallback.onFailure()。

传递的错误代码如下：

- **UNSUPPORTED_DEVICE**：设备不是 Samsung 设备或不支持 S Pen Remote。
- **CONNECTION_FAILED**： S Pen Framework 拒绝连接。
- **UNKOWN**：未知错误。

2.3 断开连接

如果您的应用程序已停止或无法再处理 S Pen 事件，请调用 disconnect()来终止已建立的连接。

3. 使用 SpenUnitManager

SpenUnitManager 提供了设置事件监听器的方法来监听 S Pen 中嵌入单元的事件。

3.1 相关的类和接口

- **SpenUnit**
  此类管理特定 S Pen 嵌入式单元的实例。
- **SpenEvent**
  S Pen Framework 将发送该类到您的应用程序，来传递 S Pen 事件信息。
- **SpenEventListener**
  您可以使用此接口创建一个 Callback 方法

3.2 监控 S Pen 事件

要监听 S Pen 事件，需要执行以下步骤：

1） 获取 SpenUnitManager 的实例

如前所述，当成功连接到 S Pen Framework 时，可以获得 SpenUnitManager 实例。

2） 请求要监视的嵌入式单元（SpenUnit）的实例

使用 getUnit 方法获取特定嵌入式单元的实例。

getUnit 方法接受常量，**TYPE_BUTTON** 或 **TYPE_AIR_MOTION**。如果尝试在设备中获取不受支持的单元实例，则返回 **null**。

```
// get Instance of Button Unit
SpenUnit button = mSpenUnitManager.getUnit(SpenUnit.TYPE_BUTTON);

// get Instance of AirMotion Unit
SpenUnit airMotion = mSpenUnitManager.getUnit(SpenUnit.TYPE_AIR_MOTION);
```

3） 注册事件监听器

使用 SPenUnitManager 实例的 registerSpenEventListener 方法注册 SpenUnit 的事件监听器。

传递给 SpenEventListener 的 SpenEvent 数据是隐藏的，您可以将其转换为 ButtonEvent 或 AirMotionEvent 类来进行处理。

- **按钮事件：**

```
SpenUnit button = mSpenUnitManager.getUnit(SpenUnit.TYPE_BUTTON);
mSpenUnitManager.registerSpenEventListener(mButtonEventListener, button);
...
//EventListener for Button Unit
private SpenEventListener mButtonEventListener = new SpenEventListener() {
    @Override
    public void onEventChanged(SpenEvent ev) {
        ButtonEvent buttonEvent = new ButtonEvent(ev)

        switch (buttonEvent.getAction()) {
            case ButtonEvent.ACTION_DOWN:
                Log.d(TAG, "Spen Button Pressed");
                break;
            case ButtonEvent.ACTION_UP:
                Log.d(TAG, "Spen Button Released");
                break;
        }
    }
};
```

- **AirMotion 事件:**

```
SpenUnit airMotion = mSpenUnitManager.getUnit(SpenUnit.TYPE_AIR_MOTION);
mSpenUnitManager.registerSpenEventListener(mAirMotionEventListener, airMotion);
...
//EventListener for AirMotion Unit
private SpenEventListener mAirMotionEventListener = new SpenEventListener() {
    @Override
    public void onEventChanged(SpenEvent ev) {
        AirMotionEvent airMotionEvent = new AirMotionEvent(ev)

        float deltaX = airMotion.getDeltaX();
        float deltaY = airMotion.getDeltaY();

        Log.d(TAG, "Air Motion = " + deltaX + ", " + deltaY);
    }
};
```

4) 取消注册事件监听器

如果您的应用程序已终止或不再处理 S Pen 事件，请使用 SPenUnitManager 实例的 unregisterSpenEventListener 方法取消事件监听器。特别是，AirMotion 消耗大量的 S Pen 电池电量，因此在不处理 Air Motion 事件时应取消事件监听器。

```
SpenUnit airMotion = mSpenUnitManager.getUnit(SpenUnit.TYPE_AIR_MOTION);
mSpenUnitManager.unregisterSpenEventListener(airMotion);
```

4. 联系我们

如果您对 S Pen Remote SDK 有任何问题，可以发邮件到下面的邮箱与我们联系。
邮箱地址：rdtpservice@samsung.com
邮件主题：三星 S Pen+APP 名