Python游戏手柄事件读取

作者: 阿凯爱玩机器人 | QQ: 244561792 | 微信: xingshunkai

1Z实验室 | B站 | 知乎

概要

本文介绍了如何使用Python读取游戏手柄的事件。 基于`input`库实现。

软硬件版本

- **操作系统**: `Ubuntu 20.04`

  > 注: Windows更简单一些，代码都是通用的。

- **手柄型号**: `谷粒金刚2 Pro`

  手柄与PC通过TypeC数据线链接, 功能模式切换到Windows D

  > 注: D代表直连。

查看USB设备信息

在Linux下执行命令行, 查看USB设备信息

```
sudo cat /sys/kernel/debug/usb/devices
```

输出日志

> 注: 日志只呈现了手柄相关的信息

```
...
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 23 Spd=12   MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=0079 ProdID=0122 Rev= 1.09
S:  Manufacturer=ZhiXu
S:  Product=GuliKit Controller D
C:* #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=400mA
I:* If#= 0 Alt= 0 #EPs= 2 Cls=03(HID  ) Sub=00 Prot=00 Driver=usbhid
E:  Ad=81(I) Atr=03(Int.) MxPS=  64 Ivl=5ms
E:  Ad=01(O) Atr=03(Int.) MxPS=  64 Ivl=10ms
...
```

于是我们可以知道手柄的`Vendor`与`ProdID`的编号.

```
Vendor=0079 ProdID=0122
```

注:ID号为16进制，所以应该写为

```
Vendor=0x0079 ProdID=0x0122
```

开发环境配置

用户组配置

对于Linux操作系统，需要先将当前用户添加到`inputs group`组里面.

查看当前用户组

```
$ groups
kyle adm dialout cdrom sudo dip plugdev lpadmin lxd sambashare docker
```

将当前用户`kyle`添加到用户组`dialout`里面

```
sudo usermod -a -G dialout kyle
```

添加完成之后，系统需要重启。

安装inputs

`inputs` 是一个输入设备读取的库。 相比较其他的库， inputs的依赖最小，最简洁。

PyPi | 使用文档
安装`inputs`

```
sudo pip3 install inputs
```

`inputs`使用实例

获取设备列表

源码

```
# 获取所有的USB设备
from inputs import devices
# 遍历所有设备名称，并打印出来
for device in devices:
    print(device)
```

日志

```
SINO WEALTH Gaming KB
USB OPTICAL MOUSE
SINO WEALTH Gaming KB  Mouse
ZhiXu GuliKit Controller D
SINO WEALTH Gaming KB  System Control
```

其中`ZhiXu GuliKit Controller D`是谷粒金刚2 Pro的设备名称.

获取游戏手柄事件并打印

```
from inputs import get_gamepad
# 打印
while True:
    # 获取游戏手柄事件
    events = get_gamepad()
    # 逐一打印事件信息
    for event in events:
        # 数据类型 |字符串|字符串|整数|
        print(f"事件类型: {event.ev_type} | 事件代码: {event.code} | 事件状态:
{event.state}")  # 字符串
    print("-"*50)
```

操作游戏手柄，查看事件输出。

> 注: 每款手柄的功能按键事件可能有出入， 以实际为准。

游戏手柄事件

遍历所有的按键操作，得到日志输出。 有了这些日志输出， 就可以进行二次开发跟封装了。

遥杆

遥杆X,Y的坐标值取值范围为$[0, 255]$, 默认为$127/128$。

左侧遥杆 `ABS_X` , `ABS_Y`

```
事件类型: Absolute | 事件代码: ABS_X | 事件状态: 119
--------------------------------------------------
事件类型: Absolute | 事件代码: ABS_Y | 事件状态: 138
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

右侧遥杆 `ABS_Z`, `ABS_RZ`

```
事件类型: Absolute | 事件代码: ABS_Z | 事件状态: 240
--------------------------------------------------
事件类型: Absolute | 事件代码: ABS_RZ | 事件状态: 124
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

右侧功能按键

按键按下会连续发送三个事件， 同样按键抬起也会触发三个事件。

状态代表按键状态 按下: `1` / 抬起: `0`

```
# 哪个按键
事件类型: Misc
事件代码: MSC_SCAN
事件状态: {按键/事件ID}
# 按键状态
事件类型: Key
事件代码: {按键名称}
事件状态: {按键状态} # 状态代表按键状态 按下: 1 / 抬起: 0
# 完成 同步
# 也有可能是多个按键按下,  然后再同步
事件类型: Sync
事件代码: SYN_REPORT
事件状态: 0
```

按键X(上)-`BTN_TOP`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589828
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TOP | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589828
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TOP | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键B(下)-`BTN_TRIGGER`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589825
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TRIGGER | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589825
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TRIGGER | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键Y(左)-`BTN_THUMB2`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589827
--------------------------------------------------
事件类型: Key | 事件代码: BTN_THUMB2 | 事件状态: 1
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589827
--------------------------------------------------
事件类型: Key | 事件代码: BTN_THUMB2 | 事件状态: 0
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键A(右)-`BTN_THUMB`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589826
--------------------------------------------------
事件类型: Key | 事件代码: BTN_THUMB | 事件状态: 1
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589826
--------------------------------------------------
事件类型: Key | 事件代码: BTN_THUMB | 事件状态: 0
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

左侧十字按键

按键上-`ABS_HAT0Y`

按键按下

```
事件类型: Absolute | 事件代码: ABS_HAT0Y | 事件状态: -1
--------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Absolute | 事件代码: ABS_HAT0Y | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键下-`ABS_HAT0Y`

> 注: 按键上跟按键下使用的按键代码都是一样的，只是按键按下的时候，事件状态一个是`-1`一个是`1`

按键按下

```
事件类型: Absolute | 事件代码: ABS_HAT0Y | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Absolute | 事件代码: ABS_HAT0Y | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键左-`ABS_HAT0X`

按键按下

```
事件类型: Absolute | 事件代码: ABS_HAT0X | 事件状态: -1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Absolute | 事件代码: ABS_HAT0X | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键右-`ABS_HAT0X`

> 注: 按键左跟按键右使用的按键代码都是一样的，只是按键按下的时候，事件状态一个是`-1`一个是`1`

按键按下

```
事件类型: Absolute | 事件代码: ABS_HAT0X | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Absolute | 事件代码: ABS_HAT0X | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

右侧顶部按键

顶部按键与右侧功能键的逻辑类似

按键R1-`BTN_PINKIE`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589830
-------------------------------------------------
事件类型: Key | 事件代码: BTN_PINKIE | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589830
-------------------------------------------------
事件类型: Key | 事件代码: BTN_PINKIE | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键R2-`BTN_BASE2`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589832
-------------------------------------------------
事件类型: Key | 事件代码: BTN_BASE2 | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589832
-------------------------------------------------
事件类型: Key | 事件代码: BTN_BASE2 | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

左侧顶部按键

按键L1-`BTN_TOP2`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589829
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TOP2 | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键抬起

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589829
-------------------------------------------------
事件类型: Key | 事件代码: BTN_TOP2 | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键L2-`BTN_BASE`

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589831
-------------------------------------------------
事件类型: Key | 事件代码: BTN_BASE | 事件状态: 1
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

按键按下

```
事件类型: Misc | 事件代码: MSC_SCAN | 事件状态: 589831
-------------------------------------------------
事件类型: Key | 事件代码: BTN_BASE | 事件状态: 0
-------------------------------------------------
事件类型: Sync | 事件代码: SYN_REPORT | 事件状态: 0
```

联系作者

深感机器人

成都深感机器人科技有限责任公司
Chengdu DeepSense Robotics Technology Co.,Ltd

邢顺凯 ShunKai Xing

机器人工程师 **Robot Engineer**

扫码加微信好友

☎ (86) 13285816609
244561792

xingshunkai@qq.com
www.1zlab.com