定制自己的游戏手柄类

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

1Z实验室 | B站 | 知乎

创建手柄库

定义一个自己的游戏手柄类， 例如`gulikit.py`。 根据测试得到的事件信息， 配置手柄的传感器。

> 详情见: **游戏手柄事件测试数据**

```
from gamepad import GamePad
from sensor.button import Button
from sensor.cross_button import CrossButtonSingle, \
    CrossButtonSingleValue, CrossButton
from sensor.joystick import JoyStickAxis, JoyStick

class Gulikit(GamePad):
    '''谷粒金刚游戏手柄
    测试机型: 谷粒金刚2Pro
    '''
    def init_sensor(self):
        '''传感器初始化'''
        # GamePad 构成元素
        # 右侧功能按键
```

```
        self.btn_a = Button("A", "Key", "BTN_THUMB", \
            logging_level=self.logging_level)
        self.btn_b = Button("B", "Key", "BTN_TRIGGER", \
            logging_level=self.logging_level)
        self.btn_x = Button("X", "Key", "BTN_TOP", \
            logging_level=self.logging_level)
        self.btn_y = Button("Y", "Key", "BTN_THUMB2", \
            logging_level=self.logging_level)
        # 右侧顶部按键
        self.btn_r1 = Button("R1", "Key", "BTN_PINKIE", \
            logging_level=self.logging_level)
        self.btn_r2 = Button("R2", "Key", "BTN_BASE2", \
            logging_level=self.logging_level)
        # 左侧顶部按键
        self.btn_l1 = Button("L1", "Key", "BTN_TOP2", \
            logging_level=self.logging_level)
        self.btn_l2 = Button("L2", "Key", "BTN_BASE", \
            logging_level=self.logging_level)
        # 左侧十字按键
        self.btn_up = CrossButtonSingle("UP", "Absolute", "ABS_HAT0Y", \
            button_click_value=CrossButtonSingleValue.BUTTON_CLICK1, \
            logging_level=self.logging_level)
        self.btn_down = CrossButtonSingle("DOWN", "Absolute", "ABS_HAT0Y", \
            button_click_value=CrossButtonSingleValue.BUTTON_CLICK2, \
            logging_level=self.logging_level)
        self.btn_left = CrossButtonSingle("LEFT", "Absolute", "ABS_HAT0X", \
            button_click_value=CrossButtonSingleValue.BUTTON_CLICK1, \
            logging_level=self.logging_level)
        self.btn_right = CrossButtonSingle("RIGHT", "Absolute", "ABS_HAT0X",
            button_click_value=CrossButtonSingleValue.BUTTON_CLICK2,
            logging_level=self.logging_level)
        self.btn_cross = CrossButton("CROSS_BUTTON", self.btn_up, self.btn_down,\
            self.btn_left, self.btn_right, logging_level=self.logging_level)
        # 左侧遥杆
        self.left_joystick_x = JoyStickAxis("LEFT_JOYSTICK_X", "Absolute",
"ABS_X", \
            logging_level=self.logging_level)
        self.left_joystick_y = JoyStickAxis("LEFT_JOYSTICK_Y", "Absolute",
"ABS_Y", \
            logging_level=self.logging_level)
        self.left_joystick = JoyStick("LEFT_JOYSTICK", self.left_joystick_x, \
            self.left_joystick_y, logging_level=self.logging_level)
        # 右侧遥感
        self.right_joystick_x = JoyStickAxis("RIGHT_JOYSTICK_X", "Absolute",
"ABS_Z", \
            logging_level=self.logging_level)
        self.right_joystick_y = JoyStickAxis("RIGHT_JOYSTICK_Y", "Absolute",
"ABS_RZ", \
            logging_level=self.logging_level)
        self.right_joystick = JoyStick("RIGHT_JOYSTICK", self.right_joystick_x, \
            self.right_joystick_y, logging_level=self.logging_level)
```

游戏手柄事件测试数据

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