

# GamePad-Python | 游戏手柄

---

## GamePad-Python | 游戏手柄

概要

使用手册

快速入手

联系作者

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

[1Z实验室](#) | [B站](#) | [知乎](#)



## 概要

---

在 `inputs` 库的基础上封装了一个游戏手柄库 `GamePad-Python`，主要功能如下：

- 创建了三种传感器类，包括普通按键、十字按键、摇杆。
- 根据自己的手柄布局，自由配置组合自己的游戏手柄类。
- 基于事件的，灵活的回调函数配置。例如按键事件: 按键按下，按键抬起，以及按键更新。
- 多线程手柄事件查询，可终止。

## 使用手册

---

- [inputs库使用教程](#)
- [游戏手柄按键事件\(谷粒金刚2Pro\)](#)
- [GamePad库Python使用示例](#)

## 快速入手

---

导入依赖

```
# 日志模块
import logging
# 自定义库
# 用户需要根据自己手柄的情况，编写gamepad的子类
# 在init_sensor函数里面定义都有哪些传感器组件，以及编号
# 详情参见`gamepad.py`与`gulikit.py`
from gulikit import Gulikit
```

## 创建手柄对象

```
# 创建手柄，可以设置不同的日志模式
# 在INFO等级下，可以显示手柄所有的按键交互日志信息。
pad = Gulikit(logging_level=logging.INFO)
# pad = Gulikit(logging_level=logging.WARN)
```

## 定义回调函数

### 键盘事件绑定

```
# 绑定回调函数
def a_click():
    print("按键A按下")

def a_release():
    print("按键A释放")

def a_change():
    print("按键A数值变动")

pad.btn_a.on_click = a_click
pad.btn_a.on_release = a_release
pad.btn_a.on_change = a_change
```

### 十字按键绑定

注: 十字按键的四个元素，都可以当普通按键来使用。

```
def cross_btn_update():
    position = pad.btn_cross.get_position()
    print(f"十字按键更新，坐标: {position}")

def cross_btn_release():
    print("十字按键释放")

pad.btn_cross.on_change = cross_btn_update
pad.btn_cross.on_release = cross_btn_release
```

### 摇杆按键事件绑定

```
def left_joystick_change():
    position = pad.left_joystick.get_position()
    print(f"摇杆位置更新: {position}")

pad.left_joystick.on_change = left_joystick_change
```

开启手柄事件监听线程

```
# 手动执行事件监听器  
pad.start_event_listener_thread()
```

详细信息，见手册 [GamePad库Python使用示例](#)

## 联系作者



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

☎ (86) 13285816609  
📞 244561792

✉ xingshunkai@qq.com  
💻 www.1zlab.com

扫码加微信好友

