key | Pygame中文文档

pygame.key

与键盘相关的 Pygame 模块。

函数

* pygame.key.get\_focused() — 当窗口获得键盘的输入焦点时返回 True
* pygame.key.get\_pressed() — 获取键盘上所有按键的状态
* pygame.key.get\_mods() — 检测是否有组合键被按下
* pygame.key.set\_mods() — 临时设置某些组合键为被按下状态
* pygame.key.set\_repeat() — 控制重复响应持续按下按键的时间
* pygame.key.get\_repeat() — 获取重复响应按键的参数
* pygame.key.name() — 获取按键标识符对应的名字

该模块包含处理与键盘操作相关的函数。当键盘按键被按下和释放时，事件队列将获得 pygame.KEYDOWN 和pygame.KEYUP 事件消息。这两个消息均包含 key 属性，是一个整数的 id，代表键盘上具体的某个按键。

pygame.KYEDOWN 事件还有个额外的属性 unicode 和 scancode。unicode 代表一个按键翻译后的 Unicode 编码，这包含 shift 按键和组合键。scancode 是扫描码，不同键盘间该值可能不同。不过这对于特殊按键像多媒体键的选择是有用的。

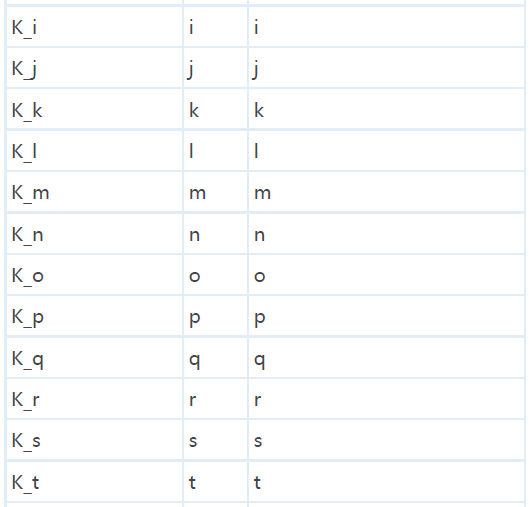
提示：当键盘按下的时候，键盘会发送一个扫描码给系统。扫描码是键盘反馈哪一个按键被按下的方式，不同类型的键盘扫描码不同。再由系统调用相应的函数将其转换为统一的 Unicode 编码。

key 属性的值是一个数字，为了方便使用，Pygame 将这些数字定义为以下这些常量：





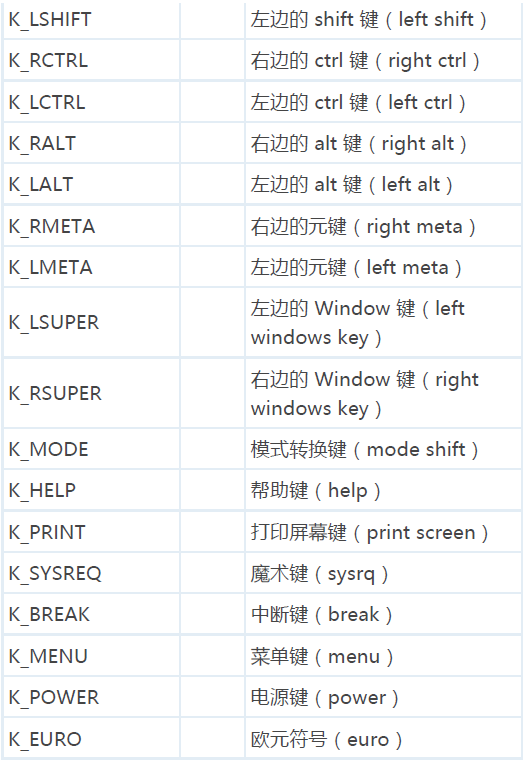












还有一个 mod 属性，用于描述组合键状态。

以下是组合键的常量定义：



提示：如果 mod & KMOD\_CTRL 是真的话，表示用户同时按下了 Ctrl 键。

函数详解

pygame.key.get\_focused()

当窗口获得键盘的输入焦点时返回 True。

*get\_focused() -> bool*

当窗口获得键盘的输入焦点时返回 True，如果窗口需要确保不失去键盘焦点，可以使用 pygame.event.set\_grab(True) 独占所有的输入接口。

提示：注意，这样做你就无法将鼠标移出窗口客户区了，但你仍然可以通过 Ctrl - Alt - Delete 热键“解围”。

pygame.key.get\_pressed()

获取键盘上所有按键的状态。

*get\_pressed() -> bools*

返回一个由布尔类型值组成的序列，表示键盘上所有按键的当前状态。使用 key 常量作为索引，如果该元素是 True，表示该按键被按下。

使用该函数获取一系列按钮被按下的状态，并不能正确的获取用户输入的文本。因为你无法知道用户按键的被按下的顺序，并且快速的连续按下键盘可能无法完全被捕获（在两次调用 pygame.key.get\_pressed() 的过程中被忽略），也无法将这些按下的按键完全转化为字符值。实现此功能可以通过捕获 pygame.KEYDOWN 事件消息来实现。

pygame.key.get\_mods()

检测是否有组合键被按下。

*get\_mods() -> int*

返回一个包含所有组合键位掩码的整数。使用位操作符 & 你可以检测某个组合键是否被按下。

提示：假如 pygame.key.get\_mods() 返回值存放在 mods 变量中，如果 mods & KMOD\_CTRL 为 True，表示 ctrl 键正被按下。

pygame.key.set\_mods()

临时设置某些组合键为被按下状态。

*set\_mods(int) -> None*

创建一个位掩码整数，包含你需要设置为被按下状态的组合键。

小甲鱼温馨提示：比如我们需要设置 ctrl 和 alt 组合键为按下状态，则可以 mods = KMOD\_CTRL | KMOD\_ALT，然后调用 pygame.key.set\_mods(mods)，这样尽管用户没有按下 ctrl 和 alt 组合键，它们依然是显示被按下状态。

pygame.key.set\_repeat()

控制重复响应持续按下按键的时间。

*set\_repeat() -> None*

*set\_repeat(delay, interval) -> None*

当开启重复响应按键，那么用户持续按下某一按键，就会不断产生同一 pygame.KEYDOWN 事件。delay 参数设置多久后（单位是毫秒）开始发送第一个 pygame.KEYDOWN 事件。interval 参数设置发送两个事件之间的间隔。如果不传入任何参数，表示取消重复响应按键。

pygame.key.get\_repeat()

获取重复响应按键的参数。

*get\_repeat() -> (delay, interval)*

当开启重复响应按键，那么用户持续按下某一按键，就会不断产生同一 pygame.KEYDOWN 事件。返回值是一个二元组，第一个元素 delay 表示多久后（单位是毫秒）开始发送第一个 pygame.KEYDOWN 事件。第二个元素 interval 表示发送两个事件之间的间隔。

默认情况下重复响应按键是没有开启的。

pygame.key.name()

获取按键标识符对应的名字。

*name(key) -> string*

获取一个按键标识符对应的字符串描述。