

鱼C论坛 论坛 技术交流区 Python交流 《零基础入门学习Python》 cdrom | Pygame中文文档

发帖

回复

返回列表

查看: 1043 | 回复: 2

[Pygame] cdrom | Pygame中文文档 [\[复制链接\]](#)



 发表于 2015-6-18 23:50:18 | 只看该作者

1# 电梯直达 ☐ 

小甲鱼  



累计签到: 2113 天
连续签到: 11 天

最佳答案 78

2613 主题 | 1万 帖子 | 1万 荣誉



管理员

鱼C - 小甲鱼



技术值 184

[发消息](#)

cdrom | Pygame中文文档

注: 本文档由 [MS节操](#) 翻译, 小甲鱼校对。

pygame.cdrom

Pygame 中使用音频 cdrom 的模块。

注: cdrom = Compact Disc Read-Only Memory 即只读光盘

函数

- pygame.cdrom.init() —— 初始化 cdrom 模块
- pygame.cdrom.quit() —— 还原 cdrom 模块
- pygame.cdrom.get_init() —— 如果 cdrom 模块初始化完成, 则返回 true
- pygame.cdrom.get_count() —— 返回系统中 cd 驱动器的个数

类

- pygame.cdrom.CD —— 用于管理 cdrom 驱动器的类

cdrom 模块管理计算机上的 CD 和 DVD 驱动器, 也可以控制音频光盘的播放和回放。这个组件在做任何事之前必须进行初始化。你创建的每一个 CD 对象都代表一个 cdrom 驱动器, 而且在使用其绝大部分功能前需要被一个个单独初始化。

函数详解

pygame.cdrom.init()

初始化 cdrom 组件。

init() -> None

初始化 cdrom 模块。该方法将扫描系统内所有的 CD 设备。在其他功能的方法被使用前, 这个组件必须先进行初始化。当你调用 pygame.init() 方法时, pygame.cdrom.init() 会自动被调用, 完成模块的初始化。调用此方法超过一次以上是安全的。

pygame.cdrom.quit()

还原 cdrom 模块。

quit() -> None

还原 cdrom 模块。在你调用该方法后, 现存的任何 CD 对象都将停止工作。调用此方法超过一次以上是安全的。

pygame.cdrom.get_init()

如果 cdrom 模块初始化完成,则返回 true。

get_init() -> bool

如果 cdrom 模块初始化完成则返回 true, 否则返回 false。因为每个驱动器必须单独初始化, 所以该方法不同于下面的 pygame.cdrom.CD.init()。

pygame.cdrom.get_count()

返回系统中 cd 驱动器的个数。

get_count() -> count

返回系统中 cd 驱动器的个数。当你创建一个 CD 对象的时候, 你需要传递一个低于该方法返回值的整数 ID。

类 class pygame.cdrom.CD

用于管理 cdrom 驱动器的类。

CD(id) -> CD

方法

- pygame.cdrom.CD.init() —— 初始化一个 cdrom 驱动器用于使用
- pygame.cdrom.CD.quit() —— 还原一个正在使用的 cdrom 驱动器
- pygame.cdrom.CD.get_init() —— 如果这个 CD 设备初始化完成, 返回 true
- pygame.cdrom.CD.play() —— 开始播放音频
- pygame.cdrom.CD.stop() —— 停止播放音频
- pygame.cdrom.CD.pause() —— 暂停播放音频
- pygame.cdrom.CD.resume() —— 恢复播放音频
- pygame.cdrom.CD.eject() —— 弹出或者打开 cdrom 驱动器
- pygame.cdrom.CD.get_id() —— 返回 cdrom 驱动器的 ID
- pygame.cdrom.CD.get_name() —— 返回 cdrom 驱动器的系统名称
- pygame.cdrom.CD.get_busy() —— 如果驱动器正在播放音频, 返回 true
- pygame.cdrom.CD.get_paused() —— 如果驱动器被暂停, 返回 true
- pygame.cdrom.CD.get_current() —— 返回当前音频的播放位置
- pygame.cdrom.CD.get_empty() —— 如果驱动器内存在 cdrom, 返回 false
- pygame.cdrom.CD.get_numtracks() —— 返回 cdrom 上的音轨数
- pygame.cdrom.CD.get_track_audio() —— 如果 cdrom 的音轨上有音频数据, 返回 true
- pygame.cdrom.CD.get_all() —— 获取所有音轨信息
- pygame.cdrom.CD.get_track_start() —— 返回一条音轨的开始时间
- pygame.cdrom.CD.get_track_length() —— 返回一条音轨的长度

你可以为系统内的每一个 cdrom 创建一个 CD 对象。使用 pygame.cdrom.get_count() 去确定目前现存驱动器的个数。id 参数是驱动器的一个整数, 从 0 开始编号。

如果这个 CD 对象没有被初始化, 你只可以调用 CD.get_id() 方法以及 CD.get_name() 方法。

对一个驱动器创建多个 CD 对象是安全的行为, 正常情况下这些对象会相互合作运行。

方法详解

pygame.cdrom.CD.init()

初始化一个 cdrom 驱动器用于使用。

init() -> None

初始化一个 cdrom 驱动器用于使用。为了满足大多数 CD 类方法运行的需要, 这个驱动器必须被初始化, 即使 pygame 的其他部分已经初始化了。

当这个驱动器被初始化的时候可能会有一个片刻的暂停时间。如果你的程序无法接受 1~2 秒的短暂停, 则要避免使用到 CD.init()。

pygame.cdrom.CD.quit()

还原一个正在使用的 cdrom 驱动器。

quit() -> None

还原一个正在使用的 cdrom 驱动器。当你的系统在一段时间内将不会访问驱动器的时候,调用该方法。

pygame.cdrom.CD.get_init()

如果这个 CD 设备初始化完成,返回 true。

get_init() -> bool

检验这个 CD 设备是否完成初始化工作。它和上面提到的 pygame.cdrom.init() 不同,因为每一个驱动器必须单独进行初始化工作。

请注意区分两者的区别:pygame.cdrom.init() 用于初始化整个 cdrom 模块,pygame.cdrom.CD.init() 则用于初始化每一个 CD 驱动器,所以当你仅调用了 pygame.cdrom.init() 方法后,pygame.cdrom.CD.get_init() 的返回值应该是 false 而不是 true,因为此时你并没有初始化任何一个 CD 驱动器。

pygame.cdrom.CD.play()

开始播放音频。

play(track, start=None, end=None) -> None

从某个驱动器内的音频 CD 中播放音频。除开音轨参数外,你也可以通过传递开始和结束时间来进行播放。

开始和结束时间以秒为单位,而且可以界定出被播放的音轨内的一小部分。

如果你仅仅传递了开始时间,音频将播放到当前音轨的末尾。如果你传递了确切的开始时间并且使结束时间为 None,音频将播放到整个磁盘的末尾(注意与上一条进行区分)。

通过调用 CD.get_numtracks() 和 CD.get_track_audio() 来寻找要播放的音轨。

注意:一张 CD 内音轨从 0 开始计数编号。

pygame.cdrom.CD.stop()

停止播放音频。

stop() -> None

停止音频播放。该方法会清除当前的播放位置。如果驱动器当前没有在播放音频,则该方法什么也不做。

pygame.cdrom.CD.pause()

暂停播放音频。

pause() -> None

暂时停止 CD 内的音频播放。通过调用 CD.resume() 可以在暂停的位置继续播放音频。如果驱动器当前没有在播放音频,则该方法什么也不做。

pygame.cdrom.CD.resume()

恢复播放音频。

resume() -> None

重新开始播放音频。如果驱动器当前没有在播放音频或暂停播放音频,则该方法什么也不做。

pygame.cdrom.CD.eject()

弹出或者打开 cdrom 驱动器。

eject() -> None

该方法将打开 CD 驱动器并且弹出 CD。如果驱动器正在播放或者被暂停,那么驱动器的工作会先自动被结束然后调用该方法。

pygame.cdrom.CD.get_id()

返回 cdrom 驱动器的 ID。

get_id() -> id

返回当初被用于创建此 CD 对象的整数型标志号（即 ID）。这个方法可以用于没有被初始化的 CD 对象上。

pygame.cdrom.CD.get_name()

返回 cdrom 驱动器的系统名称。

get_name() -> name

返回驱动器的名称（用字符串表示）。这个名称被用于代表对应的驱动器。通常是驱动号或者设备名。该方法可以用于未初始化的 CD 对象上。

pygame.cdrom.CD.get_busy()

如果驱动器正在播放音频，返回 true。

get_busy() -> bool

如果驱动器正在播放音频，返回 true。

pygame.cdrom.CD.get_paused()

如果驱动器被暂停，返回 true。

get_paused() -> bool

如果驱动器被暂停，返回 true。

pygame.cdrom.CD.get_current()

返回当前音频的播放位置。

get_current() -> track, seconds

返回当前音频的播放位置（用所在音轨号和音轨内对应时间点表示）。无论驱动器是正在播放还是被暂停，该方法都可以调用。

注意：CD 内音轨号是从 0 开始（不是 1）。

pygame.cdrom.CD.get_empty()

如果驱动器内存在 cdrom，返回 false。

get_empty() -> bool

如果驱动器当前存在 cdrom，则返回 false。如果驱动器当前是空的，则返回 true。

pygame.cdrom.CD.get_numtracks()

返回 cdrom 上的音轨数。

get_numtracks() -> count

返回驱动器内 CD 上的音轨数。如果驱动器内没有 CD 或者 CD 上没有音轨，则返回 0。

pygame.cdrom.CD.get_track_audio()

如果 cdrom 的音轨上有音频数据，返回 true。

get_track_audio(track) -> bool

确认 CD 上的某条音轨内是否包含音频数据。你也可以通过调用 CD.num_tracks() 和 CD.get_all() 确认更多关于 CD 的信

息。

pygame.cdrom.CD.get_all()

获取所有音轨信息。

get_all() -> [(audio, start, end, lenth), ...]

返回一个包含 CD 上每一条音轨信息的列表。每条信息都是一个四元组 (audio, start, end, lenth)。如果音轨上包含音频数据, 则 audio=true。start, end, lenth 均是以秒为单位的浮点数。start、end 代表整张磁盘上的绝对时间。

pygame.cdrom.CD.get_track_start()

返回一条音轨的开始时间。

get_track_start(track) -> seconds

返回音轨的绝对开始时间 (以秒为单位)。

注意 : CD 内的音轨从 0 开始计数编号。

pygame.cdrom.CD.get_track_length()

返回一条音轨的长度。

get_track_length(track) -> seconds

返回一条音轨的长度 (用以秒为单位的浮点数表示)。

注意 : CD 内的音轨从 0 开始计数编号。

分享到: QQ好友和群 QQ空间 腾讯微博 腾讯朋友

收藏1 评分 分享 淘帖 顶 踩



如果您的【问题求助】得到满意的解答, 请自行将分类修改为【已经解决】

回复

举报

发表于 2016-10-13 11:28:50 | 只看该作者

2#

很赞

Lyman



累计签到 : 11 天
连续签到 : 1 天

最佳答案 0

0

9

9

主题

帖子

荣誉