

pygame最小开发框架

```
#导入必须包
import pygame
from sys import exit

#常量设置
SCREENSIZE = (600,400)#窗口尺寸，格式是一个二元组，代表宽和高。
BG_COLOR = (255,255,255)#颜色，格式是一个三元组，代表RGB。

#pygame初始化
pygame.init()
pygame.display.set_caption("Your Caption")
screen = pygame.display.set_mode(SCREENSIZE)

#游戏循环
while True:
    eventList = pygame.event.get()#获得事件
    screen.fill(BG_COLOR)#用BG_COLOR填充背景
    for event in eventList:#对事件逐一响应
        if(event.type == pygame.QUIT):
            exit()
        #if(Your conditions)...:
        #    do_something()
    pygame.display.flip()#刷新屏幕
```

这里需要解释的大概有两点：

1. 关于 `pygame.display.init()`：

对这个问题感兴趣的同学可以阅读 [【pygame】pygame的init\(\) - sc0T7_ly - 博客园\(cnblogs.com\)](#)

2. 关于 `for event in eventList`：

`eventList`是在前面 `eventList = pygame.event.get()` 定义的，它是一个列表，里面包含了这次循环到上次循环之中所有的pygame所定义的事件，包括但不限于鼠标点击，键盘按下弹起，游戏退出等等等等。

而对于这个for循环，每次循环中，`event`就代表这个列表中的一个元素，也就是一个单独的事件，所以我们只需要单独的去检测这个事件是什么，并且对此作出相应的回应就可以了。

`event`有一个属性 `type`，详细对照见下表：

事件类型	成员属性
QUIT	none
ACTIVEEVENT	gain, state
KEYDOWN	unicode, key, mod
KEYUP	key, mod
MOUSEMOTION	pos, rel, buttons
MOUSEBUTTONUP	pos, button
MOUSEBUTTONDOWN	pos, button
JOYAXISMOTION	joy, axis, value
JOYBALLMOTION	joy, ball, rel
JOYHATMOTION	joy, hat, value
JOYBUTTONUP	joy, button
JOYBUTTONDOWN	joy, button
VIDEORESIZE	size, w, h
VIDEOEXPOSE	none
USEREVENT	code

所以我们首先要确定事件的类型，也就是

```
for event in eventList:
    if event.type == pygame.QUIT:#这些判断来确定类型
        exit()
```

确定好类型之后就可以对这个事件进行对应的反馈了，比如按下某个案件，那么就可以

```
for event in eventList:
    if event.type == pygame.KEYDOWN:#这些判断来确定类型
        if event.key == pygame.K_ESCAPE:#这些来确定该类型的成员
            exit()
```

来对各种不同的按键作出不同的反馈。

还有一个关于pygame.display.set_mode(SIZE,mode)，请[移步此处](#)