Pygame游戏设计

第03章 – 事件

Python编程课

事件

- 事件可以处理游戏中的各种事情。简单的说，很多程序都需要对“发生的事情”做出反应，
- 比如说：

  移动或点击鼠标；

  按键；

  经过了一定的时间；

获取事件

- Pygame会接受用户的各种操作（比如按键盘，移动鼠标等）产生的事件。事件随时可能发生，而且量也可能会很大，pygame的做法是把一系列的事件存放在一个队列里，逐个处理，我们把这个队列称为事件队列。
- 我们在之前的程序中使用了如下代码：
- ```
  for event in pygame.event.get():
      if event.type == QUIT:
          pygame.quit()
          sys.exit()
  ```
- 我们使用pygame.event.get()来获取事件，并用for循环遍历事件队列，只针对有用的事件作出处理，比如关闭窗口时产生的QUIT事件，该事件发生时，退出pygame及sys。这就是事件检索及处理。


  ```

记录事件

下面，我们来写一个程序，实现用txt文件记录所有的事件。

```
import sys
import pygame
from pygame.locals import *

pygame.init()
screen_size = width, height = 480, 700
screen = pygame.display.set_mode(screen_size)
pygame.display.set_caption( “RecordEvent” )
#创建并打开一个txt文件，用来记录事件信息
f = open( “record.txt” , ‘w’ )
while True:
    for event in pygame.event.get():    #检索所有的事件
        f.write(str(event) +  ‘\n’ )   #将事件信息转换为字符串，并写入文件
        if event.type == QUIT:
            f.close()    #当关闭窗口以后，关闭文件
            pygame.quit()
            sys.exit()
    pygame.display.flip()
```

- 鼠标移动的事件信息：
- <Event(4-MouseMotion {'pos': (351, 360), 'rel': (-16, 11), 'buttons': (0, 0, 0)})>
  <Event(4-MouseMotion {'pos': (335, 373), 'rel': (-16, 13), 'buttons': (0, 0, 0)})>

- 键盘按键按下和松开的事件信息：
- <Event(2-KeyDown {'unicode': '', 'key': 97, 'mod': 0, 'scancode': 30})>
  <Event(3-KeyUp {'key': 97, 'mod': 0, 'scancode': 30})>
  <Event(2-KeyDown {'unicode': '', 'key': 98, 'mod': 0, 'scancode': 48})>
  <Event(3-KeyUp {'key': 98, 'mod': 0, 'scancode': 48})>
  <Event(2-KeyDown {'unicode': '', 'key': 99, 'mod': 0, 'scancode': 46})>
  KeyDown表示按键被按下的事件；KeyUp表示按键松开的事件； ‘key’ 值表示按键对应字母的ASCII码，97表示A，98表示B ……

显示文字

- 有没有更酷的方法来显示所有发生的事件呢？能不能把事件信息显示在游戏窗口上呢？窗口背景还要是黑色的，显示的字要是绿色的，还能‘唰唰唰...’地显示在窗口上。
- 步骤：

  设置游戏窗口背景颜色；

  获取事件，并将事件信息显示在游戏窗口；

  当游戏窗口显示占满，重新从第一行开始显示；

- 首先，我们要先将pygame游戏的基本框架化做出来，然后将游戏窗口背景设置为黑色。

- ```
  import sys
  import pygame
  from pygame.locals import *

  pygame.init()
  bg = (0, 0, 0)   #背景rgb值设为黑色
  screen_size = width, height = 480, 700
  screen = pygame.display.set_mode(screen_size)
  pygame.display.set_caption( "displayRecord" )
  screen.fill(bg)   #使用纯黑色填充背景

  while True:
      for event in pygame.event.get():
          if event.type == QUIT:
              sys.exit()
              pygame.quit()
      pygame.display.flip()
  ```

pygame的基础框架程序，我们要自己动手再写一次哦！你会越来越熟悉它的。


  ```

pygame.font

- 游戏窗口有了，背景也设置为黑色的了。那么，该怎样显示文字呢？Pygame不能直接在窗口打印文字。但是可以通过font对象的render()方法将文字渲染成surface对象（一个图像），然后调用blit()方法将该surface对象绘制到窗口surface对象上，就可以实现显示文字的效果啦！
- pygame.font模块是pygame中加载和表示文字的模块，该模块包含一个类pygame.font.Font，用来从一个字体文件创建一个font对象。
- # filename为字体文件的名字，None则表示使用系统默认字体；size为字体大小
- Font(filename, size) -> Font
- Font(object, size) -> Font
- font对象包含很多有用的方法，比如render()方法能够创建一个新的surface对象，并在上面渲染指定的文本，该方法会返回一个新的surface对象；
- get_linesize()方法可以获取字体文本的行高。

类

- 什么是类？类和对象有什么关系？
- 猫科动物（类，就是分类的类）的特点：大多数猫科动物皮毛有斑点、花纹，头部稍大，捕猎，食肉等等。
- 老虎（对象，某一类的实例）属于猫科动物，拥有猫科动物的一些特点以及能力。

- 在python中，有一些已经存在的基础类，比如Font类。我们也可以通过class关键字自定义一个类，这个我们在后面的课程中会讲到。类中会包含所有属于这个类的对象的一些共有的特性及能力，在我们实例化一个对象时，这个对象就会具备这个类的特性及能力。

- 下面我们将显示文字的程序添加到基础框架程序上,创建字体对象->获取行高。

- ```
  import sys
  import pygame
  from pygame.locals import *

  pygame.init()
  bg = (0, 0, 0)

  screen_size = width, height = 480, 700
  screen = pygame.display.set_mode(screen_size)
  pygame.display.set_caption("displayRecord")
  screen.fill(bg)
  #文字显示
  font = pygame.font.Font(None, 20)    #使用Font类创建一个字体对象，并命名为font， None表示默认字体，20表示字体大小
  height_per_line = font.get_linesize()    #使用font对象的get_linesize()方法获取每行文字的行高（占多少像素）
  lineHeight = 0    #定义变量，表示垂直坐标的递增
  ```


  ```

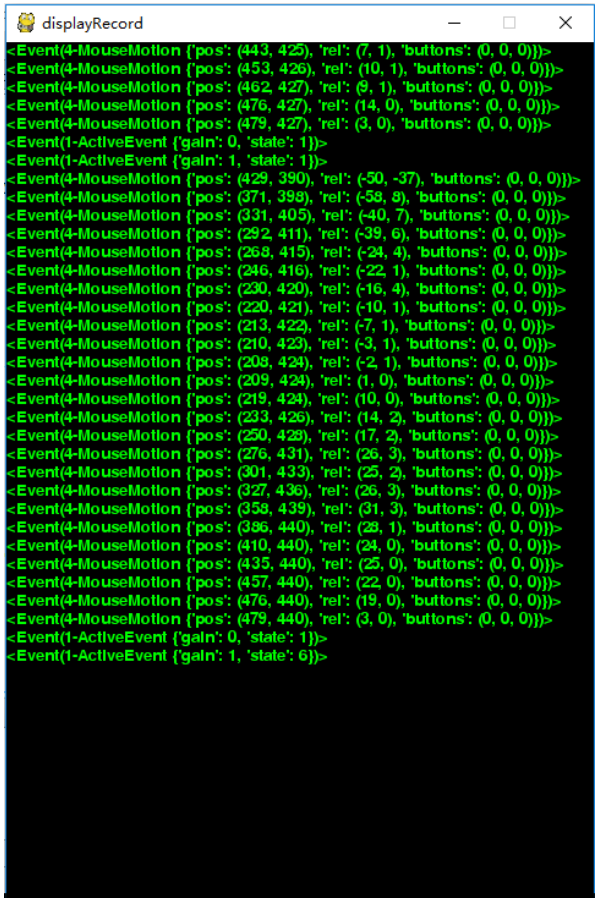
- 下面我们将显示文字的程序实现出来,创建文字surface对象绘制到窗口surface对象上。
- **while True**:

```
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
            pygame.quit()
        # 调用font对象的render()方法创建文字surface对象，
        # 并命名font_surface
        font_surface = font.render(str(event), True, (0, 255, 0))
        # 参数分别为要显示的文字，抗锯齿，字体颜色
        # 将文字surface对象（图像）绘制在窗口surface对象（图像）上
        # 从每一行的左上角的像素开始。
        screen.blit(font_surface, (0, lineHeight))
        # 每绘制完一行，行高变量自增加每一行的高度
        lineHeight += height_per_line
        # 如果行高变量大于窗口高度时，说明窗口占满，需清零行高，
        # 以及重新用黑色覆盖所有的文字
        if lineHeight > height:
            lineHeight = 0
            screen.fill(bg)

    pygame.display.flip()
```

运行结果

- 运行程序，看看结果吧！ “唰唰唰...”是不是很酷！



- 下面的表中列出了pygame中常用的事件类型，大家可以随时查阅。

事件	含义	属性
QUIT	按下关闭按钮	none
ATIVEEVENT	Pygame被激活或者隐藏	gain, state
KEYDOWN	键盘按键被按下	unicode, key, mod
KEYUP	键盘按键被松开	key, mod
MOUSEMOTION	鼠标移动	pos, rel, buttons
MOUSEBUTTONDOWN	鼠标按键被按下	pos, button
MOUSEBUTTONUP	鼠标按键被松开	pos, button
JOYAXISMOTION	游戏手柄上的摇杆移动	joy, axis, value
JOYBALLMOTION	游戏手柄上的轨迹球滚动	joy, axis, value
JOYHATMOTION	游戏手柄上的帽子开关移动	joy, axis, value
JOYBUTTONDOWN	游戏手柄按钮被按下	joy, button
JOYBUTTONUP	游戏手柄按钮被松开	joy, button
VIDEORESIZE	用户调整窗口的尺寸	size, w, h
VIDEOEXPOSE	部分窗口需要重新绘制	none
USEREVENT	用户定义的事件	code

游戏控制

- 游戏需要有互动，我们要能控制它来完成游戏的任务，下面我们来使用键盘的方向按键操控小飞机。
- 首先借助前面的程序，实现在游戏窗口的左上角显示小飞机；
- 检测键盘上的方向按键，控制小飞机往左飞，往右飞，往上飞，往下飞；
- 窗口边界检测，小飞机不能飞出窗口。

- 在游戏窗口的左上角显示小飞机。
-
```
import sys
import pygame
from pygame.locals import *

pygame.init()
bg = (0, 255, 0)
screen_size = width, height = 480, 700
screen = pygame.display.set_mode(screen_size)
pygame.display.set_caption("controlAirplane")

airPlane = pygame.image.load("me1.png")
position = airPlane.get_rect()

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
            pygame.quit()
    screen.fill(bg)
    screen.blit(airPlane,position)
    pygame.display.flip()
```

设置速度

- 下面，添加代码，我们要让小飞机动，动，动起来~
- 先初始化速度（包含移动方向）

```
pygame.init()
bg = (0, 255, 0)

#初始化移动速度，用列表表示
#第一个元素表示左右移动，第二个元素表示上下移动
speed = [0, 0]
screen_size = width, height = 480, 700
```

- 判断KEYDOWN事件，并判断哪个键被按下，修改速度speed的值，改变小飞机移动方向和速度。
- **while True**:

```
    for event in pygame.event.get():
        if event.type == QUIT:
            sys.exit()
            pygame.quit()
        if event.type ==  KEYDOWN:    #判断KEYDOWN事件
            #判断哪个按键被按下，第一个元素-1和1分别表示向左和向右移动一个像素点，
            if event.key == K_LEFT:
                speed = [-1, 0]
            if event.key == K_RIGHT:
                speed = [1, 0]
            if event.key == K_UP:
                speed = [0, -1]
            if event.key == K_DOWN:
                speed = [0, 1]
        else:
            speed = [0, 0]      #不按键的时候速度清零，小飞机不移动
```

- 更新position，移动小飞机：

- position = position.move(speed)　#使用move()方法更新position
  screen.fill(bg)
  screen.blit(airPlane,position)
  pygame.display.flip()

- 现在小飞机可以随着我们按方向键移动了，但是它会飞出去，下面我们要解决这个问题。不能让小飞机飞出游戏窗口。

- 限制小飞机飞出游戏窗口。

- #如果超出边界，那么小飞机的位置等于边界值，然后调用move()方法，更新小飞机的位置。

- ```
  if position.left < 0:
      position.left = 0
  if position.right > width:
      position.right = width
  if position.top <0:
      position.top = 0
  if position.bottom > height:
      position.bottom = height
  position = position.move(speed)
  ```


  ```