




# Python编程案例教程

# 第13章 游戏开发

## 本章导读



在Python中实现游戏开发，首选必然是pygame。pygame是一组功能强大且有趣的模块，可用于管理图形、图像、动画及声音等，使用户能够更轻松地开发复杂的游戏。本章将介绍如何使用pygame模块进行游戏开发。



# 学习目标

- 掌握pygame的安装方法
- 掌握pygame模块的基本使用方法
- 掌握面向对象编程思路
- 掌握使用Python编写大型程序的方法

The background of the slide features an abstract design composed of overlapping, semi-transparent red polygons of various shades, creating a dynamic, geometric pattern on the left side.

**13.1 pygame简介**

**13.2 pygame常用功能**

**13.3 典型案例——飞机大战**

The background of the slide is an abstract composition of various shades of red and pink. It features a complex pattern of overlapping triangles and polygons, creating a sense of depth and movement. The colors range from deep, dark reds to lighter, almost white pinks, with some areas showing a gradient effect. The overall style is modern and geometric.

## 13.1 pygame简介

## 13.1 pygame简介

- ◆ pygame是一组用来开发游戏的Python程序模块，它是在SDL库的基础上开发的。
- ◆ 用户可以利用它，在Python程序中创建功能丰富的游戏和多媒体程序。
- ◆ pygame是一组高可移植性的模块，可以支持多个操作系统，非常适合小游戏的开发。

### 提示

SDL ( Simple DirectMedia Layer ) 是一套开放源代码的跨平台多媒体开发库，使用C语言编写而成。SDL提供了数种控制图形、图像、声音、输入和输出的函数，使用户只要用相同或相似的代码就可以开发出跨多个平台 ( Linux , Windows , Mac OS X等 ) 的应用软件。目前SDL多用于开发游戏、模拟器、媒体播放器等多媒体应用领域。



## 13.1 pygame简介

由于pygame不是Python默认集成的模块，因此，需要通过pip3指令进行安装，pip3安装命令如下：

```
C:\>pip3 install pygame
```

安装完成后，输入以下命令，如果出现如下图所示示例图，则说明安装成功。

```
C:\>python -m pygame.examples.aliens
```



### 提示

如果pip3指令安装不成功，可登录  
<http://www.pygame.org/download.shtml>  
网站，找到对应Python版本的pygame模块，  
下载之后直接默认安装即可。



## 13.2 pygame常用功能

- ◆ 13.2.1 创建窗口并显示图片
- ◆ 13.2.2 绘制图形
- ◆ 13.2.3 动画效果
- ◆ 13.2.4 播放声音
- ◆ 13.2.5 显示文字
- ◆ 13.2.6 事件



### 13.2.1 创建窗口并显示图片

pygame中最重要的就是surface对象，它是pygame用来表示图像的对象。

#### ► 1 . pygame.display.set\_mode()方法

pygame.display.set\_mode()方法用于初始化一个窗口或屏幕。其方法原型如下：

`set_mode(resolution=(0,0), flags=0, depth=0)`

表示使用的颜色深度

指定扩展选项

是一个二元组，表示宽和高

选 项	含 义
FULLSCREEN	创建一个全屏显示
DOUBLEBUF	双缓冲模式（推荐与HWSURFACE或OPENGL一起使用）
HWSURFACE	硬件加速，只有在FULLSCREEN下可以使用
OPENGL	创建一个OPENGL渲染的显示
RESIZABLE	创建一个可调整尺寸的窗口
NOFRAME	创建一个没有边框和控制按钮的窗口

## 13.2.1 创建窗口并显示图片

### ► 2 . pygame.image.load()方法

pygame.image.load()方法用于加载一张新的图片。其方法原型如下：

```
load(filename)
```

为要加载的图片的名称（可包含路径）

使用pygame.image.load()可创建一个包含图像的surface对象。创建完surface对象后，要想显示图像，还需要调用blit()方法，该方法用于将一个图像（surface对象）绘制到另一个图像上方。其方法原型如下：

```
blit(source, dest[, area=None, special_flags = 0])
```

为坐标值，一般为source的左上角坐标

为要加载的图片的名称（可包含路径）

## 13.2.1 创建窗口并显示图片

例：创建一个窗口，并加载一张图片作为窗口的背景。

#导入所需模块

```
import pygame, sys
```

#导入所有pygame.locals里的变量（比如下面大写的QUIT变量）

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

#初始化pygame

```
pygame.init()
```

#设置窗口大小，单位为像素

```
screen = pygame.display.set_mode((500, 400))
```

#创建一个背景图片

```
background = pygame.image.load("./pic/back.png")
```

#设置窗口标题

```
pygame.display.set_caption('Hello World')
```

#程序主循环

```
while True:
```

#把背景复制到窗口的(0,0)处贴进去

```
screen.blit(background, (0,0))
```

#获取事件

```
for event in pygame.event.get():
```

#判断事件是否为退出事件

```
    if event.type == QUIT:
```

```
        pygame.quit()           #退出pygame
```

```
        sys.exit()             #退出系统
```

```
    pygame.display.update()     #绘制屏幕内容
```

## 13.2.1 创建窗口并显示图片

程序运行效果



提示

如果想要全屏显示游戏界面，可在调用`pygame.display.set_mode()`方法时设置第二个参数为`FULLSCREEN`，同时加上硬件加速`HWSURFACE`，即：

```
screen = pygame.display.set_mode((500, 400), FULLSCREEN | HWSURFACE)
```

## 13.2.2 绘制图形

pygame中绘制图形的模块是pygame.draw，该模块中常用的方法如下表所示。

方 法	作 用
pygame.draw.line(surface, color, start_pos, end_pos, width)	绘制一条线段
pygame.draw.aaline(surface, color, start_pos, end_pos, blend)	绘制一条抗锯齿的线
pygame.draw.lines(surface, color, closed, pointlist, width)	绘制一条折线
pygame.draw.rect(surface, color, Rect)	绘制一个矩形
pygame.draw.ellipse(surface, color, Rect)	绘制一个椭圆
pygame.draw.ellipse(surface, color, Rect, width)	绘制一个椭圆框
pygame.draw.polygon(surface, color, pointlist, width)	绘制一个多边形
pygame.draw.arc(surface, color, Rect, start_angle, stop_angle, width)	绘制一条弧线
pygame.draw.circle(surface, color, Rect, radius)	绘制一个圆

- ◆ pygame.draw方法的第一个参数总是`surface`（即surface对象名）；
- ◆ 接下来是参数`color`，表示颜色；
- ◆ 再后就是一系列坐标等。

## 13.2.2 绘制图形

例：绘制图形实例。

```
#导入需要的模块
import pygame, sys
from pygame.locals import *
from math import pi

#初始化pygame
pygame.init()

#设置窗口大小，单位为像素
screen = pygame.display.set_mode((400,300))

#设置窗口标题
pygame.display.set_caption('Drawing')

#定义颜色
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
```

```
#设置背景颜色
screen.fill(WHITE)

#绘制一条线
pygame.draw.line(screen, RED, [50, 50], [250, 50], 2)

#绘制一条抗锯齿的线
pygame.draw.aaline(screen, GREEN, [0, 50], [50, 50], True)

#绘制一条折线
pygame.draw.lines(screen, BLACK, False, [[0, 80], [50, 80], [50, 120]], 5)

#绘制一个空心矩形
pygame.draw.rect(screen, BLACK, [75, 10, 50, 20], 2)

#绘制一个矩形
pygame.draw.rect(screen, BLACK, [150, 10, 50, 20])

#绘制一个空心椭圆
pygame.draw.ellipse(screen, RED, [225, 10, 50, 20], 2)

#绘制一个椭圆
pygame.draw.ellipse(screen, RED, [300, 10, 50, 20])
```

提示

fill()方法的作用是使用  
纯色填充surface对象。

## 13.2.2 绘制图形

#绘制多边形（三角形）

```
pygame.draw.polygon(screen, BLACK, [[120, 150], [0, 200], [200, 200]], 5)
```

#绘制多条弧线

```
pygame.draw.arc(screen, BLACK, [210, 75, 150, 125], 0, pi/2, 2)
```

```
pygame.draw.arc(screen, GREEN, [210, 75, 150, 125], pi/2, pi, 2)
```

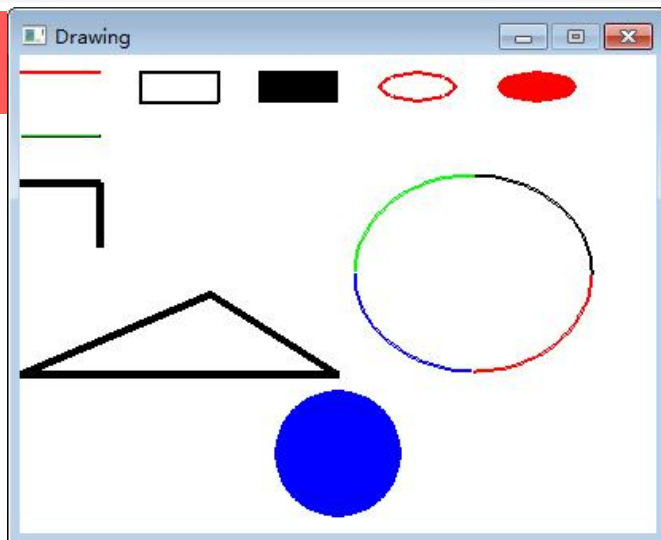
```
pygame.draw.arc(screen, BLUE, [210, 75, 150, 125], pi, 3*pi/2, 2)
```

```
pygame.draw.arc(screen, RED, [210, 75, 150, 125], 3*pi/2, 2*pi, 2)
```

#绘制一个圆

```
pygame.draw.circle(screen, BLUE, [200, 250], 40)
```

程序运行效果



#程序主循环

```
while True:
```

#获取事件

```
for event in pygame.event.get():
```

#判断事件是否为退出事件

```
if event.type == QUIT:
```

```
    pygame.quit()
```

```
    sys.exit()
```

```
pygame.display.update()
```

#退出pygame

#退出系统

#绘制屏幕内容



## 13.2.3 动画效果

- ◆ 物体移动的原理很简单，即在原有坐标的基础上增加偏移量，再重新绘制图片，依次一张一张地循环绘制下去，就可以得到物体移动的效果。
- ◆ 还有一个问题就是两个画面之间的时间间隔，即经过多长时间绘制下一张图片。
- ◆ pygame.time模块提供了一个Clock对象，调用其tick()方法即可解决上述问题：
  - ( 1 ) pygame.time.Clock()用于实例化对象以获得pygame的时钟。
  - ( 2 ) 调用pygame.time.Clock.tick(FPS)方法设置pygame时钟的间隔时间，其中参数FPS即为帧率（帧率即为每秒钟显示的帧数）。



## 13.2.3 动画效果

例：动画效果实例。

```
#导入需要的模块
import pygame, sys
from pygame.locals import *
#初始化pygame
pygame.init()
# 设置帧率（屏幕每秒刷新的次数）
FPS = 30
#获得pygame的时钟
fpsClock = pygame.time.Clock()
#设置窗口大小
screen = pygame.display.set_mode((500, 400),
0, 32)
#设置标题
pygame.display.set_caption('Animation')
#定义颜色
WHITE = (255, 255, 255)
```

```
#加载一张图片
img = pygame.image.load('./pic/yundong.png')
#初始化图片的位置
imgx = 10
imgy = 10
#初始化图片的移动方向
direction = 'right'
#程序主循环
while True:
    #每次都要重新绘制背景白色
    screen.fill(WHITE)
    #判断移动的方向，并对相应的坐标作加减
    if direction == 'right':
        imgx += 5
        if imgx == 380:
            direction = 'down'
```

## 13.2.3 动画效果

```
elif direction == 'down':  
    imgy += 5  
    if imgy == 300:  
        direction = 'left'  
elif direction == 'left':  
    imgx -= 5  
    if imgx == 10:  
        direction = 'up'  
elif direction == 'up':  
    imgy -= 5  
    if imgy == 10:  
        direction = 'right'
```

程序运行效果



```
#该方法用于将图片绘制到相应的坐标中  
screen.blit(img, (imgx, imgy))  
for event in pygame.event.get():  
    if event.type == QUIT:  
        pygame.quit()  
        sys.exit()  
#刷新屏幕  
pygame.display.update()  
#设置pygame时钟的间隔时间  
fpsClock.tick(FPS)
```

## 13.2.4 播放声音

对于一般游戏来说，声音分为音效和背景音乐两种。

- ◆ 音效是在某种条件下被触发播放的，音效可采用无压缩的wav格式。
- ◆ 而背景音乐则是时刻伴随着游戏存在的，背景音乐可采用ogg格式或者mp3格式。

### ▶ 1. 播放音效

播放音效使用mixer模块，需要首先生成一个Sound对象，其具体语法格式如下：

`pygame.mixer.Sound(filename)`

filename为音频文件名（可包含路径）

方 法	含 义	方 法	含 义
play()	播放音效	get_volume()	获取音量
stop()	停止播放	get_num_channels()	计算该音效播放的次数
fadeout()	淡出	get_length()	获取音效的长度
set_volume()	设置音量	get_raw()	将音效以二进制格式返回

## 13.2.4 播放声音

### ► 2 . 播放背景音乐

播放背景音乐使用music模块，music模块是mixer模块中的一个特殊实现，可使用pygame.mixer.music来调用该模块下的方法。

music模块支持的常用方法如下表所示：

方 法	含 义	方 法	含 义
load()	载入音乐	get_volume()	获取音量
play()	播放音乐	get_busy()	检查是否正在播放音乐
rewind()	重新播放音乐	set_pos()	设置播放的位置
stop()	结束音乐播放	get_pos()	获取播放的位置
pause()	暂停音乐播放	queue()	将一个音乐文件放入待播放队列中
unpause()	恢复音乐播放	set_endevent()	当播放结束时发出一个事件
fadeout()	淡出	get_endevent()	获取播放结束时发送的事件
set_volume()	设置音量		

## 13.2.4 播放声音

例：播放声音实例。

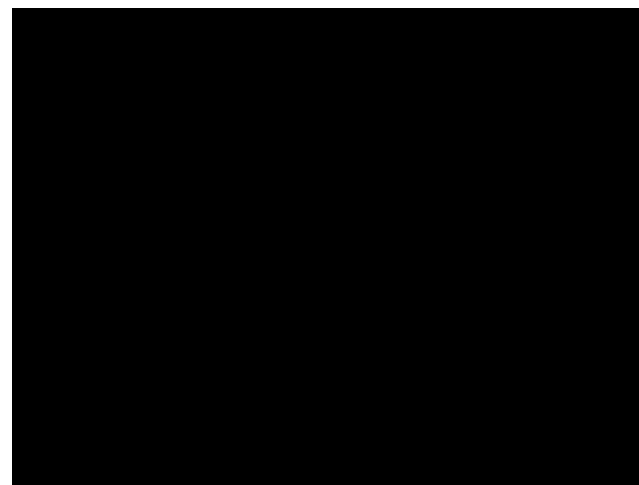
```
#导入需要的模块
import pygame, sys
from pygame.locals import *
#初始化pygame
pygame.init()
#设置窗口大小，单位为像素
screen = pygame.display.set_mode((200,150))
#设置窗口的标题
pygame.display.set_caption('Audio')
#定义颜色
WHITE = (255, 255, 255)
#设置背景颜色
screen.fill(WHITE)
#加载并播放一个特效音频文件
sound =
pygame.mixer.Sound('sound/bird.wav')
```

```
sound.play()
#加载背景音乐文件
pygame.mixer.music.load('sound/Remember.mp3')
#播放背景音乐，第一个参数为播放的次数（-1表示无限循环），第
二个参数是设置播放的起点（单位为秒）
pygame.mixer.music.play(-1, 0.0)
#程序主循环
```

## 13.2.4 播放声音

程序运行效果

```
while True:
    #获取事件
    for event in pygame.event.get():
        #判断事件是否为退出事件
        if event.type == QUIT:
            #停止播放背景音乐
            pygame.mixer.music.stop()
            #退出pygame
            pygame.quit()
            #退出系统
            sys.exit()
    #绘制屏幕内容
    pygame.display.update()
```





## 13.2.5 显示文字

如果想要将文字显示到屏幕上，可首先使用pygame.font.Font()创建一个Font对象，其语法格式如下：

```
pygame.font.Font(filename, size)
```

字体的大小，单位为像素

字体文件的文件名（一般为“.ttf”格式的文件）

创建Font对象后，就可以使用render()方法绘制文本了，其语法格式如下：

```
pygame.font.Font.render(text, antialias, color, background=None)
```

背景颜色

字体颜色

是个布尔值，决定是否开启抗锯齿，当为True时字体比较平滑

要显示的文字

最后需要调用blit()方法显示文字。

## 13.2.5 显示文字

例：显示文字实例。

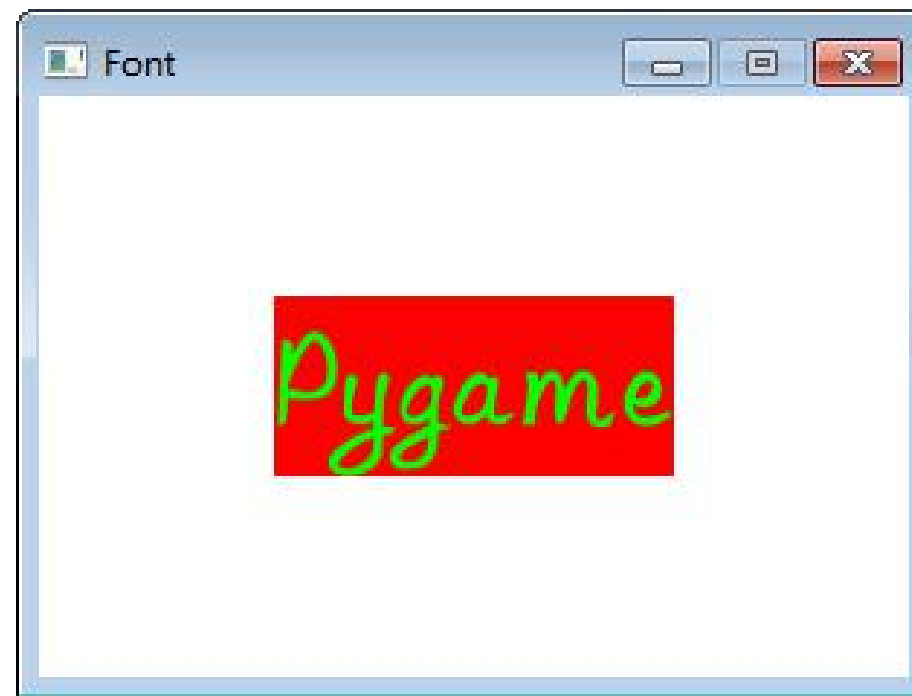
```
#导入需要的模块
import pygame, sys
from pygame.locals import *
#初始化pygame
pygame.init()
#设置窗口大小，单位为像素
screen = pygame.display.set_mode((300,200))
#设置窗口标题
pygame.display.set_caption('Font')
#定义颜色
WHITE = (255, 255, 255)
RED = ( 255, 0, 0)
GREEN = ( 0, 255, 0)
#通过字体文件获得字体对象
fontObj = pygame.font.Font('ziti/FZQTFW.TTF', 50)
```

```
#配置要显示的文字
textSurfaceObj = fontObj.render('Pygame', True, GREEN, RED)
#获得要显示的对象rect
textRectObj = textSurfaceObj.get_rect()
#设置显示对象的坐标
textRectObj.center = (150, 100)
#设置背景
screen.fill(WHITE)
#绘制字体
screen.blit(textSurfaceObj, textRectObj)
```

## 13.2.5 显示文字

```
#程序主循环
while True:
    #获取事件
    for event in pygame.event.get():
        #判断事件是否为退出事件
        if event.type == QUIT:
            #退出pygame
            pygame.quit()
            #退出系统
            sys.exit()
    #绘制屏幕内容
    pygame.display.update()
```

程序运行效果



提示

可通过`get_rect()`方法获得一个对象的结构，以便于设置其坐标位置。

## 13.2.6 事件

- ◆ 事件是pygame提供的干预机制。
- ◆ pygame会接收用户的各种操作（如敲击键盘、移动鼠标、单击鼠标等）产生事件。
- ◆ 一般可使用pygame.event.get()方法来处理所有事件，pygame中常用的事件及含义如下表所示：

事 件	含 义	属 性
QUIT	用户按下关闭按钮	none
ACTIVEEVENT	Pygame被激活或者隐藏	gain , state
KEYDOWN	键盘被按下	unicode , key , mod
KEYUP	键盘被放开	key , mod
MOUSEMOTION	鼠标移动	pos , rel , buttons
MOUSEBUTTONDOWN	鼠标按下	pos , button
MOUSEBUTTONUP	鼠标放开	pos , button
VIDEORESIZE	pygame窗口缩放	size , w , h

## 13.2.6 事件

例：事件实例。

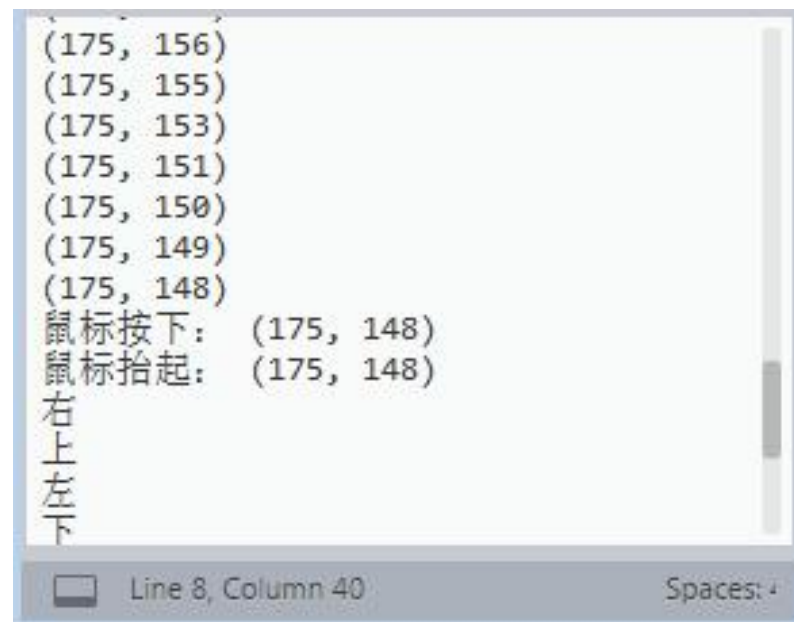
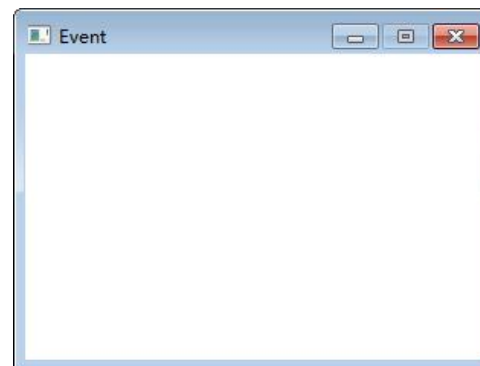
```
#导入需要的模块
import pygame, sys
from pygame.locals import *
#定义颜色
WHITE = (255, 255, 255)
#初始化pygame
pygame.init()
#设置窗口大小，单位为像素
screen = pygame.display.set_mode((300,200), 0, 32)
#设置窗口标题
pygame.display.set_caption('Event')
#设置背景
screen.fill(WHITE)
```

```
#程序主循环
while True:
    #获取事件
    for event in pygame.event.get():
        #判断事件是否为退出事件
        if event.type == QUIT:
            #退出pygame
            pygame.quit()
            #退出系统
            sys.exit()
        #获得鼠标当前位置
        if event.type == MOUSEMOTION:
            print(event.pos)
        #获得鼠标按下的位置
        if event.type == MOUSEBUTTONDOWN:
            print("鼠标按下：",event.pos)
```

## 13.2.6 事件

```
#获得鼠标抬起的位置
if event.type == MOUSEBUTTONUP:
    print("鼠标抬起：",event.pos)
#获得键盘按下的事件
if event.type == KEYDOWN:
    if(event.key==K_UP):
        print("上")
    if(event.key==K_DOWN):
        print("下")
    if(event.key==K_LEFT):
        print("左")
    if(event.key==K_RIGHT):
        print("右")
#按下键盘的Esc键退出
if(event.key==K_ESCAPE):
    #退出pygame
    pygame.quit()
    #退出系统
    sys.exit()
#绘制屏幕内容
pygame.display.update()
```

### 程序运行效果



**感谢您的观看**

