

# Pygame事件处理机制

---



嵩 天  
北京理工大学





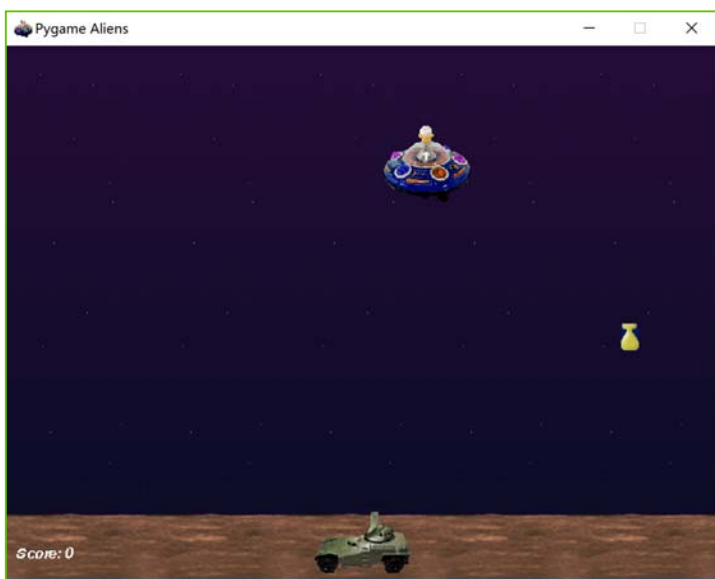
# Pygame事件处理机制





# Pygame事件处理机制简介

# 事件处理需求



- 响应用户键盘、鼠标等外设操作
- 响应屏幕尺寸和模式变化
- 响应游戏情节的特定触发条件
- 产生一些触发条件
- .....

# Pygame事件队列

```
while True:  
    for event in pygame.event.get():  
        if event.type == ...
```

用户对事件  
逐一处理



- 缓存并派发所有事件
- 原则上先到先处理



# Pygame事件

## pygame.event.EventType

- 事件本质上是一种封装后的数据类型（对象）
- EventType是Pygame的一个类，表示事件类型
- 事件类型只有属性，没有方法
- 用户可自定义新的事件类型

非使用事件队列的事件处理方式将不在本课程中介绍

# 事件类型及属性

QUIT	系统	none
ACTIVEEVENT		gain, state
KEYDOWN	键盘	unicode, key, mod
KEYUP		key, mod
MOUSEMOTION	鼠标	pos, rel, buttons
MOUSEBUTTONUP		pos, button
MOUSEBUTTONDOWN		pos, button
JOYAXISMOTION	游戏杆	joy, axis, value
JOYBALLMOTION		joy, ball, rel
JOYHATMOTION		joy, hat, value
JOYBUTTONUP		joy, button
JOYBUTTONDOWN		joy, button
VIDEORESIZE	窗口	size, w, h
VIDEOEXPOSE		none
USEREVENT	用户定义	code

实例：键盘落下事件及属性

`pygame.event.KEYDOWN`

- `event.unicode`
- `event.key`
- `event.mod`



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

# 事件处理的重要函数

`pygame.event.get()`

`pygame.event.poll()`

`pygame.event.clear()`

处理事件

`pygame.event.set_blocked()`

`pygame.event.get_blocked()`

`pygame.event.set_allowed()`

操作事件队列

`pygame.event.post()`

`pygame.event.Event()`

生成事件

更多事件处理函数将不在本课程中介绍



# 键盘事件及类型的使用

# 键盘事件及属性

`pygame.event.KEYDOWN`

键盘按下事件

- `event.unicode` 按键的unicode码，平台有关，不推荐使用
- `event.key` 按键的常量名称
- `event.mod` 按键修饰符的组合值

`pygame.event.KEYUP`

键盘释放事件

- `event.key`
- `event.mod`

# 按键的常量名称

K_BACKSPACE	K_DOLLAR	K_0	K_COLON	K_UNDERSCORE
K_TAB	K_AMPERSAND	K_1	K_SEMICOLON	K_BACKQUOTE
K_CLEAR	K_QUOTE	K_2	K_LESS	K_DELETE
K_RETURN	K_LEFTPAREN	K_3	K_EQUALS	K_UP
K_PAUSE	K_RIGHTPAREN	K_4	K_GREATER	K_DOWN
K_ESCAPE	K_ASTERISK	K_5	K_QUESTION	K_RIGHT
K_SPACE	K_PLUS	K_6	K_AT	K_LEFT
K_EXCLAIM	K_COMMA	K_7	K_LEFTBRACKET	K_INSERT
K_QUOTEDBL	K_MINUS	K_8	K_BACKSLASH	K_HOME
K_HASH	K_PERIOD	K_9	K_RIGHTBRACKET	K_END
	K_SLASH		K_CARET	

# 按键的常量名称

K_KP0	K_KP_PERIOD	K_F1	K_F11	K_CAPSLOCK	K_LSUPER
K_KP1	K_KP_DIVIDE	K_F2	K_F12	K_SCROLLOCK	K_RSUPER
K_KP2	K_KP_MULTIPLY	K_F3	K_F13	K_RSHIFT	K_MODE
K_KP3	K_KP_MINUS	K_F4	K_F14	K_LSHIFT	K_HELP
K_KP4	K_KP_PLUS	K_F5	K_F15	K_RCTRL	K_PRINT
K_KP5	K_KP_ENTER	K_F6		K_LCTRL	K_SYSREQ
K_KP6	K_KP_EQUALS	K_F7		K_RALT	K_BREAK
K_KP7	K_PAGEUP	K_F8		K_LALT	K_MENU
K_KP8	K_PAGEDOWN	K_F9		K_RMETA	K_POWER
K_KP9	K_NUMLOCK	K_F10		K_LMETA	K_EURO

# 按键的常量名称

K_a	K_j	K_s
K_b	K_k	K_t
K_c	K_l	K_u
K_d	K_m	K_v
K_e	K_n	K_w
K_f	K_o	K_x
K_g	K_p	K_y
K_h	K_q	K_z
K_i	K_r	

# 按键的修饰符

KMOD\_NONE

KMOD\_LSHIFT

KMOD\_RSHIFT

KMOD\_SHIFT

KMOD\_CAPS

KMOD\_LCTRL

KMOD\_RCTRL

KMOD\_CTRL

KMOD\_LALT

KMOD\_RALT

KMOD\_ALT

KMOD\_LMETA

KMOD\_RMETA

KMOD\_META

KMOD\_NUM

KMOD\_MODE



# 按键的修饰符

- `event.mod`

修饰符的按位或运算

```
event.mod = KMOD_ALT | KMOD_SHIFT
```

```
# Unit PYG04: Pygame Event Print
import pygame, sys

pygame.init()
screen = pygame.display.set_mode((600, 400))
pygame.display.set_caption("Pygame事件处理")

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.unicode == " ":
                print("#", event.key, event.mod)
            else:
                print(event.unicode, event.key, event.mod)

    pygame.display.update()
```

测试一下键盘的事件吧

PYG04-PygameEventPrint.py



# 鼠标事件及类型的使用

# 鼠标事件及属性

`pygame.event.MOUSEMOTION`

鼠标移动事件

`pygame.event.MOUSEBUTTONUP`

鼠标键释放事件

`pygame.event.MOUSEBUTTONDOWN`

鼠标键按下事件

# 鼠标事件及属性

`pygame.event.MOUSEMOTION`

鼠标移动事件

- `event.pos` 鼠标当前坐标值(x,y)，相对于窗口左上角
- `event.rel` 鼠标相对运动距离(x,y)，相对于上次事件
- `event.buttons` 鼠标按钮状态(a,b,c)，对应于鼠标的三个键

# 鼠标事件及属性

`pygame.event.MOUSEBUTTONUP`

鼠标键释放事件

- `event.pos` 鼠标当前坐标值(x,y)，相对于窗口左上角
- `event.button` 鼠标按下键编号n  
取值 0/1/2，分别对应三个键

# 鼠标事件及属性

`pygame.event.MOUSEBUTTONDOWN`

鼠标键按下事件

- `event.pos` 鼠标当前坐标值(x,y)，相对于窗口左上角
- `event.button` 鼠标按下键编号n  
取值为整数，左键为1，右键为3，设备相关

```
# Unit PYG04: Pygame Event Print v1
import pygame, sys

pygame.init()
screen = pygame.display.set_mode((600, 400))
pygame.display.set_caption("Pygame事件处理")
```

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        elif event.type == pygame.MOUSEMOTION:
            print("[MOUSEMOTION]:", event.pos, event.rel, event.buttons)
        elif event.type == pygame.MOUSEBUTTONUP:
            print("[MOUSEBUTTONUP]:", event.pos, event.button)
        elif event.type == pygame.MOUSEBUTTONDOWN:
            print("[MOUSEBUTTONDOWN]:", event.pos, event.button)
```

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

测试一下鼠标的事件吧

PYG04-PygameEventPrint.py





# 壁球小游戏(鼠标型)

# 壁球小游戏(鼠标型)

**需求：**

鼠标可以通过左键摆放壁球，当释放按键时壁球运动

**从需求到实现的关键要素：**

- **鼠标事件：**对于鼠标键按下和释放的合理处置

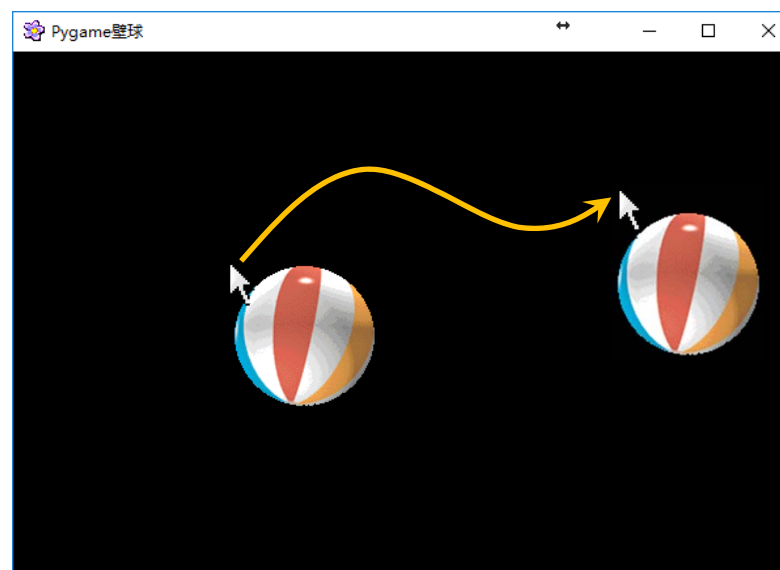
# 壁球小游戏(鼠标型)

## 鼠标左键按下

左键按下拖拽小球

## 鼠标左键释放

左键释放小球继续运动



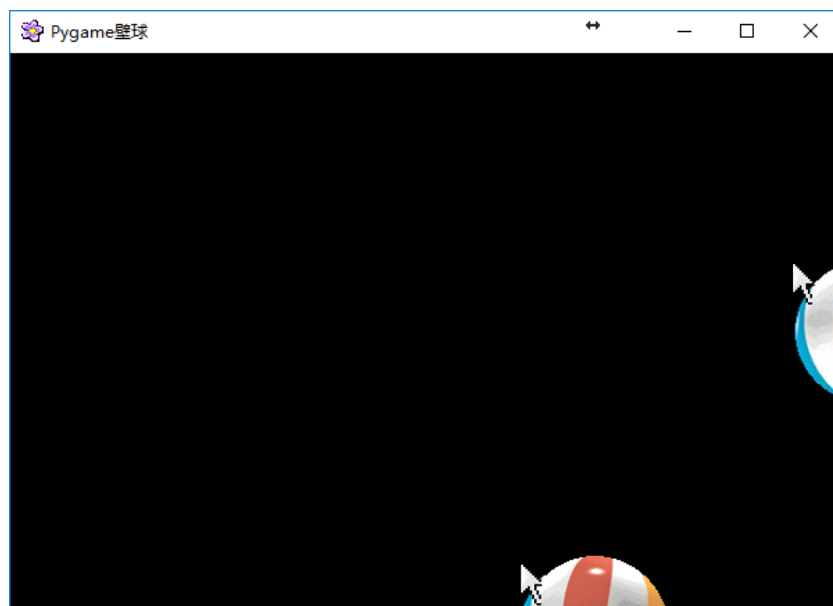
# 壁球小游戏(鼠标型)

## 局部处理

增加一种交互方式要考虑

额外的局部处理

游戏逻辑的调试



```

# Unit PYG03: Pygame Wall Ball Game version 8 鼠标型
import pygame, sys

pygame.init()
size = width, height = 600, 400
speed = [1,1]
BLACK = 0, 0, 0
screen = pygame.display.set_mode(size, pygame.RESIZABLE) #窗口大小可调

icon = pygame.image.load("PYG03-flower.png")
pygame.display.set_icon(icon)
pygame.display.set_caption("Pygame壁球")
ball = pygame.image.load("PYG02-ball.gif")
ballrect = ball.get_rect()
fps = 300
fclock = pygame.time.Clock()
still = False

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                speed[0] = speed[0] if speed[0] == 0 else (abs(speed[0]) - 1)*int(speed[0]/abs(speed[0]))
            elif event.key == pygame.K_RIGHT:
                speed[0] = speed[0] + 1 if speed[0] > 0 else speed[0] - 1
            elif event.key == pygame.K_UP:
                speed[1] = speed[1] + 1 if speed[1] > 0 else speed[1] - 1
            elif event.key == pygame.K_DOWN:
                speed[1] = speed[1] if speed[1] == 0 else (abs(speed[1]) - 1)*int(speed[1]/abs(speed[1]))
            elif event.key == pygame.K_ESCAPE:
                sys.exit()
        elif event.type == pygame.VIDEORESIZE:
            size = width, height = event.size[0], event.size[1]
            screen = pygame.display.set_mode(size, pygame.RESIZABLE)
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                still = True
        elif event.type == pygame.MOUSEBUTTONUP:
            still = False
            if event.button == 1:
                ballrect = ballrect.move(event.pos[0] - ballrect.left, event.pos[1] - ballrect.top)
        elif event.type == pygame.MOUSEMOTION:
            if event.buttons[0] == 1:
                ballrect = ballrect.move(event.pos[0] - ballrect.left, event.pos[1] - ballrect.top)
    if pygame.display.get_active() and not still:
        ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = - speed[0]
        if ballrect.right > width and ballrect.right + speed[0] > ballrect.right:
            speed[0] = - speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = - speed[1]
        if ballrect.bottom > height and ballrect.bottom + speed[1] > ballrect.bottom:
            speed[1] = - speed[1]

    screen.fill(BLACK)
    screen.blit(ball, ballrect)
    pygame.display.update()
    fclock.tick(fps)

```

# 与老师一起改改壁球小游戏(鼠标型)吧

## PYG04-PygameWallBallv8.py



# Pygame事件处理函数

# 事件处理的重要函数

`pygame.event.get()`

`pygame.event.poll()`

`pygame.event.clear()`

处理事件

`pygame.event.set_blocked()`

`pygame.event.get_blocked()`

`pygame.event.set_allowed()`

操作事件队列

`pygame.event.post()`

`pygame.event.Event()`

生成事件

更多事件处理函数将不在本课程中介绍

# 事件处理函数

## pygame.event.get()

- 从事件队列中获得事件列表，即获得所有被队列的事件

```
for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
        sys.exit()
```

- 可以增加参数，获得某类或某些类事件：

`pygame.event.get(type)`

`pygame.event.get(typelist)`



# 事件处理函数

`pygame.event.poll()`

- 从事件队列中获得一个事件

```
While True:  
    event = pygame.event.poll()
```

- 事件获取将从事件队列中删除
- 如果事件队列为空，则返回 `event.NOEVENT`

# 事件处理函数

`pygame.event.clear()`

- 从事件队列中删除事件，默认删除所有事件
- 该函数与`pygame.event.get()`类似，区别仅是不对事件进行处理
- 可以增加参数，删除某类或某些类事件：

`pygame.event.clear(type)`

`pygame.event.clear(typelist)`

# Pygame事件队列



`pygame.event.set_blocked()`

`pygame.event.set_allowed()`

`pygame.event.get_blocked()`

同时仅能存储128个事件  
当队列满时，更多事件将被丢弃

- 设置事件队列能够缓存事件的类型

# 事件处理函数

`pygame.event.set_blocked(type or typelist)`

- 控制哪些类型事件不允许被保存到事件队列中

`pygame.event.set_allowed(type or typelist)`

- 控制哪些类型事件允许被保存到事件队列中

# 事件处理函数

`pygame.event.get_blocked(type)`

- 测试某个事件类型是否被事件队列所禁止
- 如果事件类型被禁止，则返回True，否则返回False

# 事件处理函数

`pygame.event.post(Event)`

- 产生一个事件，并将其放入事件队列
- 一般用于放置用户自定义事件（`pygame.USEREVENT`）
- 也可以用于放置系统定义事件（如鼠标或键盘等），给定参数

# 事件处理函数

`pygame.event.Event(type, dict)`

- 创建一个给定类型的事件
- 其中，事件的属性和值采用字典类型复制，属性名采用字符串形式
- 如果创建已有事件，属性需要一致

```
# Unit PYG04: Pygame Event Post
```

```
import pygame, sys
```

```
pygame.init()
```

```
screen = pygame.display.set_mode((600, 400))
```

```
pygame.display.set_caption("Pygame事件处理")
```

```
fps = 1
```

```
fclock = pygame.time.Clock()
```

```
num = 1
```

```
while True:
```

```
    uevent = pygame.event.Event(pygame.KEYDOWN, {"unicode": 123, "key": pygame.K_SPACE, "mod": pygame.KMOD_ALT
```

```
    pygame.event.post(uevent)
```

```
    num = num + 1
```

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

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

```
            sys.exit()
```

```
        elif event.type == pygame.KEYDOWN:
```

```
            if event.unicode == "":
```

```
                print("[KEYDOWN {}]:".format(num), "#", event.key, event.mod)
```

```
            else:
```

```
                print("[KEYDOWN {}]:".format(num), event.unicode, event.key, event.mod)
```

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

```
fclock.tick(fps)
```

与老师一起测试一下事件产生机制吧

PYG04-PygameEventPost.py

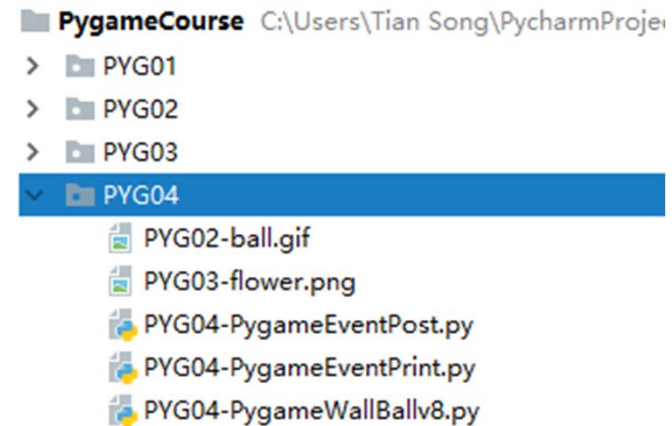




# 单元小结



- Pygame事件处理机制简介
- 键盘事件及类型的使用
- 鼠标事件及类型的使用
- 壁球小游戏(鼠标型)
- Pygame事件处理函数



## Pygame事件处理机制