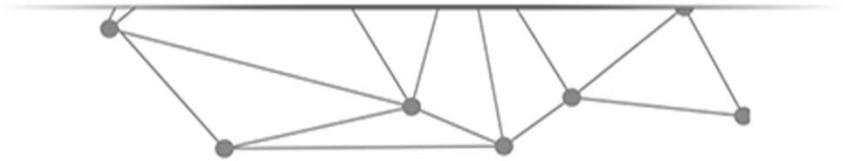
Pygame屏幕绘制机制

嵩 天
北京理工大学

单元开篇



Pygame屏幕绘制机制

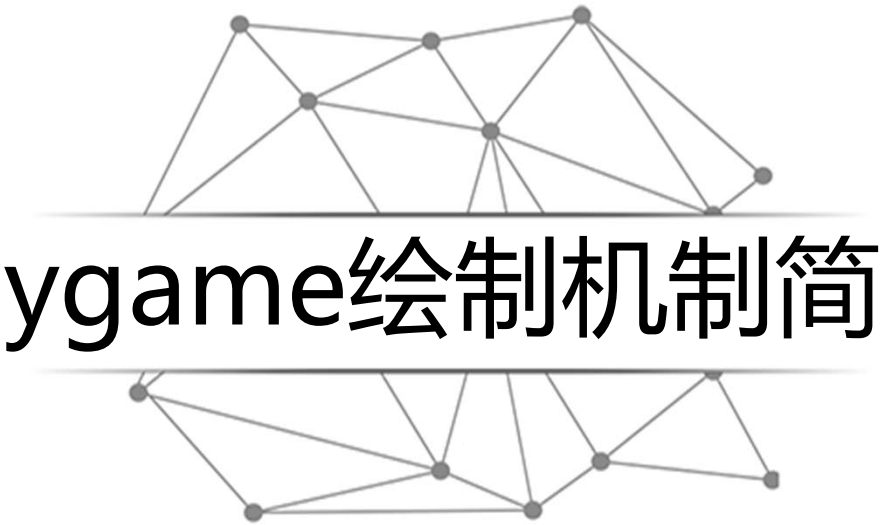
Pygame屏幕绘制机制简介

Pygame屏幕尺寸和模式设置

Pygame窗口标题和图标设置

Pygame窗口感知和刷新运用

python
弹指之间 · 享受创新



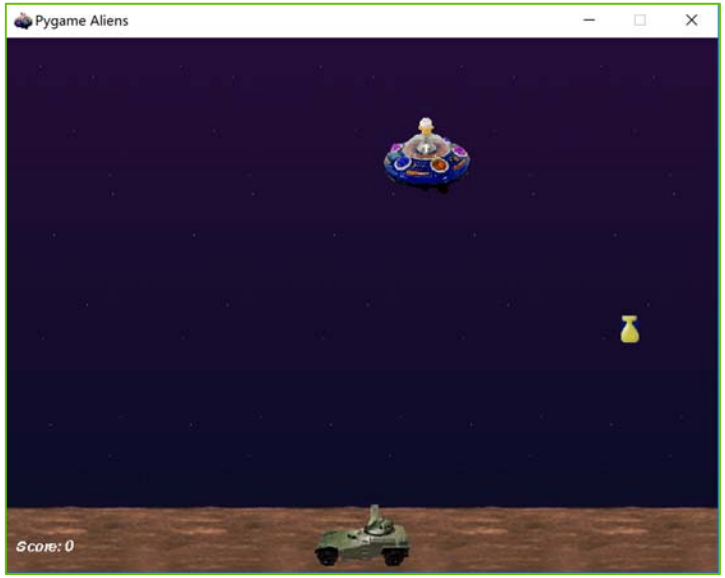
Pygame绘制机制简介

屏幕控制

pygame.display

- 用来控制Pygame游戏的屏幕
- Pygame有且只有一个屏幕
- 屏幕左上角坐标为(0,0)
- 以像素为单位

屏幕控制需求



- 游戏全屏
- 游戏屏幕大小可调节
- 游戏屏幕无边框
- 更改游戏标题栏内容
- 更改游戏图标
- ……

屏幕控制的重要函数

```
pygame.display.set_mode()
pygame.display.Info()
```

屏幕尺寸和模式

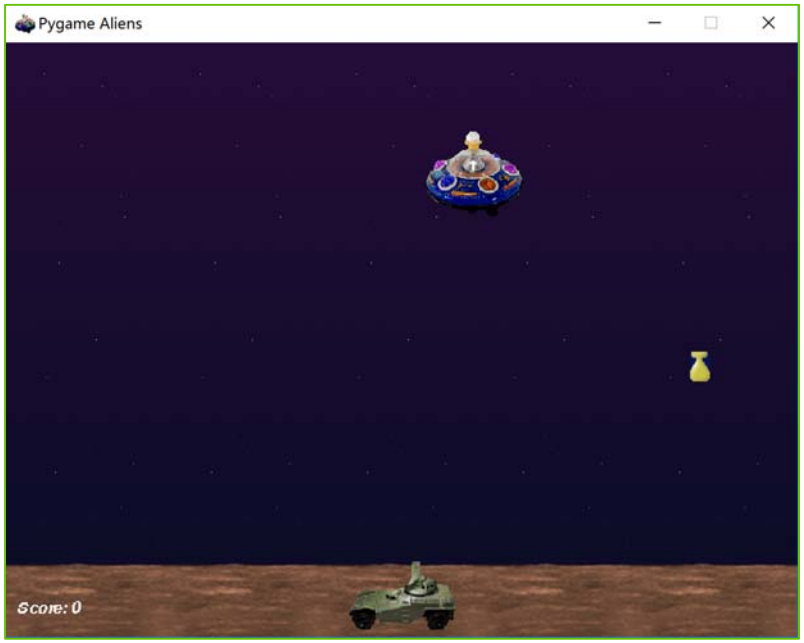
```
pygame.display.set_caption()
pygame.display.set_icon()
pygame.display.get_caption()
```

窗口标题和图标

```
pygame.display.get_active()
pygame.display.flip()
pygame.display.update()
```

窗口感知和刷新

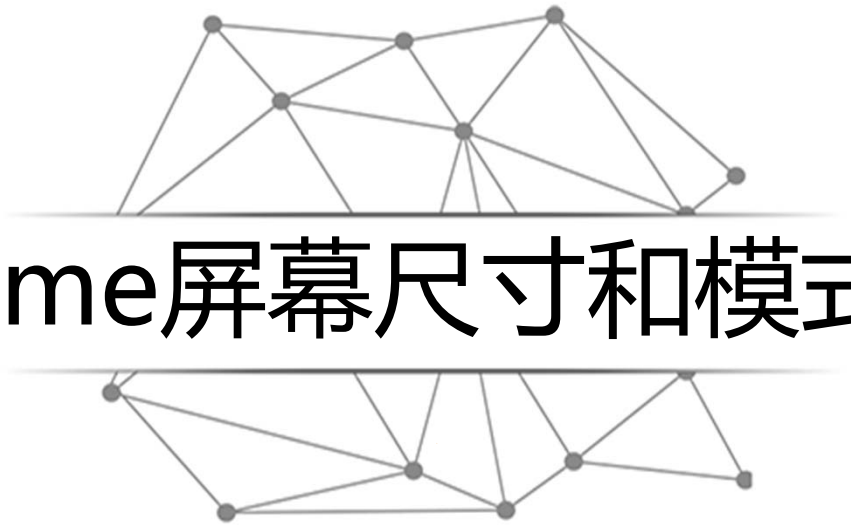
OpenGL和硬件加速



`pygame.display`

- 可以采用OpenGL支持显示
- 可以采用硬件加速显示
- 绘制加速将带来更流畅的运行效果

游戏的硬件加速将不在本课程中介绍



Pygame屏幕尺寸和模式设置

屏幕模式函数

```
pygame.display.set_mode(r=(0,0), flags=0)
```

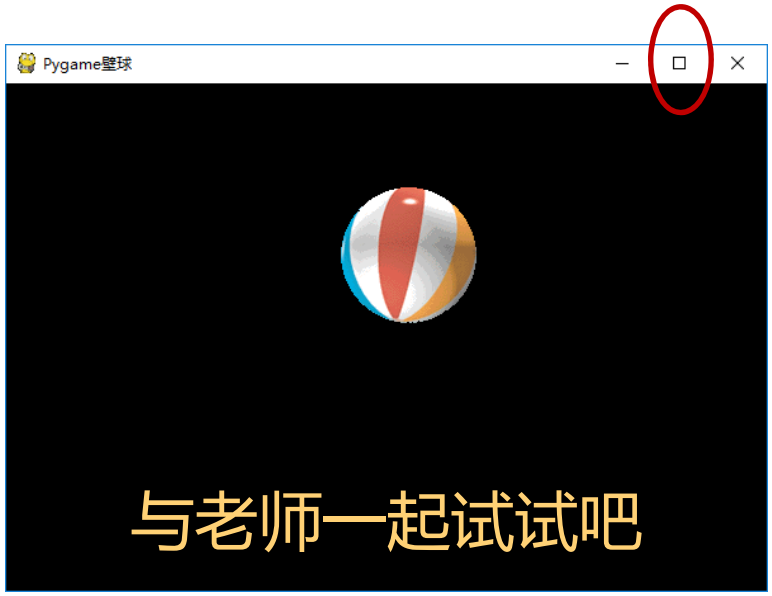
- r是游戏屏幕的分辨率，采用(width, height)方式输入
- flags用来控制显示类型，可用|组合使用，常用显示标签如下：
  - pygame.RESIZABLE　　窗口大小可调
  - pygame.NOFRAME　　窗口没有边界显示
  - pygame.FULLSCREEN　　窗口全屏显示

注意：每种显示方式要配合相应的处理机制

屏幕设置为大小可调

```
pygame.display.set_mode(r=(0,0), flags=0)
```

```
……
BLACK = 0, 0, 0
screen = pygame.display.set_mode(size, pygame.RESIZABLE)
pygame.display.set_caption("Pygame壁球")
……
```



注意：大小可调时要有尺寸变化的响应

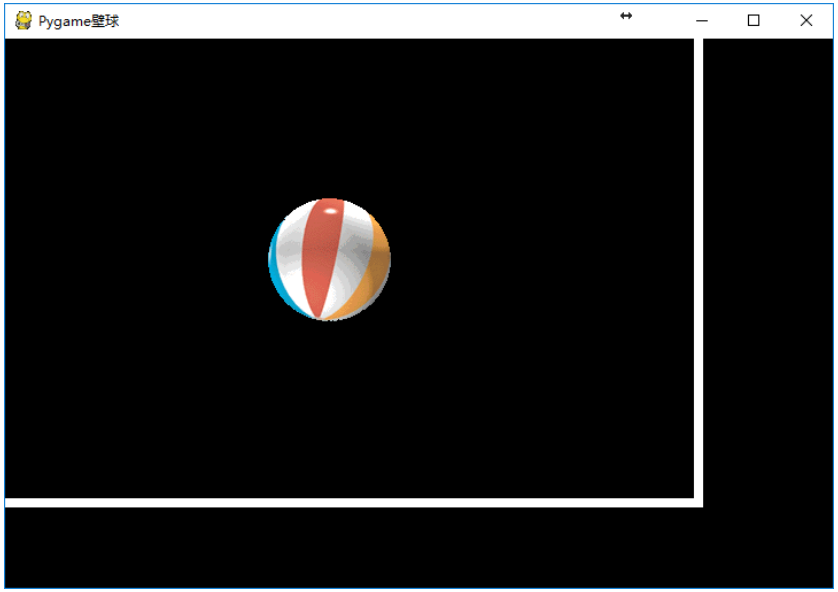
屏幕设置为大小可调

```
pygame.display.set_mode(r=(0,0), flags=0)
```

注意：

大小可调时要有尺寸变化的响应

- 对扩张显示界面的刷新
- 对扩张/缩小显示界面的游戏响应



屏幕设置为无边框

```
pygame.display.set_mode(r=(0,0), flags=0)
```

```
……
BLACK = 0, 0, 0
screen = pygame.display.set_mode(size, pygame.NOFRAME)
pygame.display.set_caption("Pygame壁球")
……
```

注意：无边框时要增加其他退出方式

屏幕设置为全屏

```
pygame.display.set_mode(r=(0,0), flags=0)
```

```
……
BLACK = 0, 0, 0
screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
pygame.display.set_caption("Pygame壁球")
……
```

注意：全屏时要考虑系统分辨率，否则将会按照用户设定被拉伸

屏幕信息函数

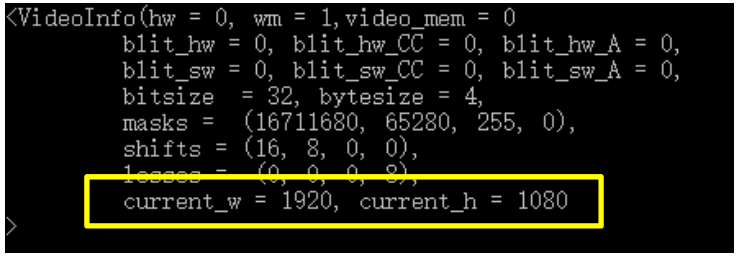
pygame.display.Info()

- 产生一个显示信息对象VideoInfo，表达当前屏幕的参数信息
- 在.set_mode()之前调用，则显示当前系统显示参数信息
- 参数很多，其中有两个十分重要，如下：
    - current_w:当前显示模式或窗口的像素宽度
    - current_h:当前显示模式或窗口的像素高度

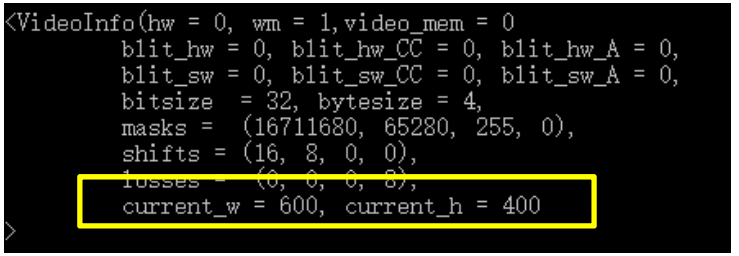
屏幕控制的重要函数

pygame.display.Info()

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



调用.set_mode()前

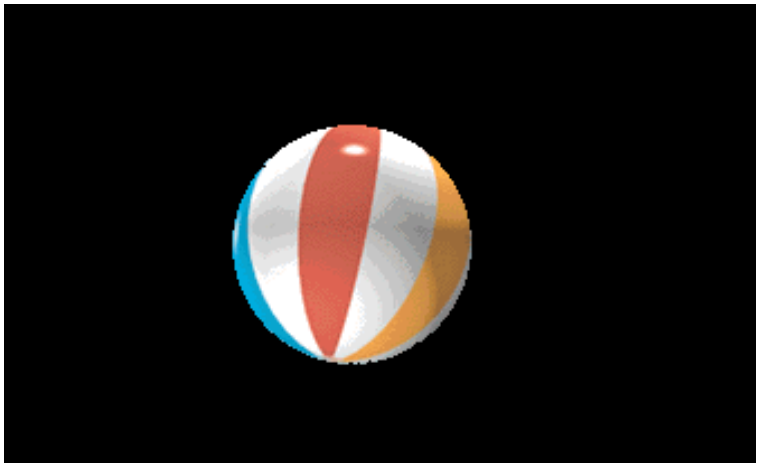


调用.set_mode()后

壁球小游戏(全屏型)

```
pygame.display.set_mode(r=(0,0), flags=0)
```

```
……
size = width, height = vInfo.current_w, vInfo.current_h
……
screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
pygame.display.set_caption("Pygame壁球")
……
```



注意：采用系统分辨率进行全屏

```
# Unit PYG03: Pygame Wall Ball Game version 4
import pygame,sys

pygame.init()
vInfo = pygame.display.Info()
size = width, height = vInfo.current_w, vInfo.current_h
speed = [1,1]
BLACK = 0, 0, 0
#screen = pygame.display.set_mode(size, pygame.RESIZABLE)
#screen = pygame.display.set_mode(size, pygame.NOFRAME)
#screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
screen = pygame.display.set_mode(size, pygame.FULLSCREEN)
pygame.display.set_caption("Pygame壁球")
ball = pygame.image.load("PYG02-ball.gif")
ballrect = ball.get_rect()
fps = 300
fclock = pygame.time.Clock()

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()
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = - speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = - speed[1]

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

看看全屏的壁球小游戏代码

（ESC键可以退出哦）

与老师一起改改壁球小游戏(全屏型)吧

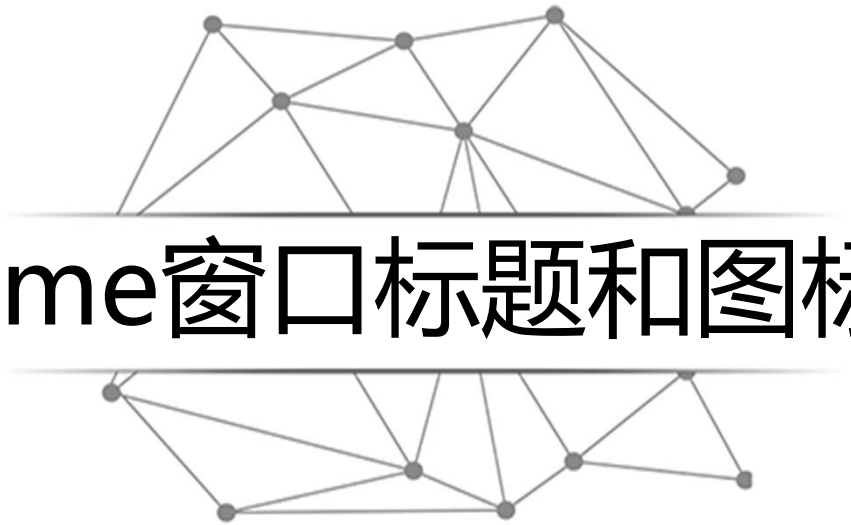
壁球小游戏(伸缩型)

Pygame.VIDEORESIZE

- 这是一种窗口大小更改的事件
- 事件发生后，返回event.size元组，包含新窗口的宽度和高度
  - .size[0] 宽度，也可以用event.w
  - .size[1] 高度，也可以用event.h
  - 返回参数仅在事件发生时有用

```
# Unit PYG03: Pygame Wall Ball Game version 5  伸缩版
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
#screen = pygame.display.set_mode(size, pygame.NOFRAME)  #窗口无边框
#screen = pygame.display.set_mode(size, pygame.FULLSCREEN)  #窗口全屏显示

pygame.display.set_caption("Pygame壁球")
ball = pygame.image.load("PYG02-ball.gif")
ballrect = ball.get_rect()
fps = 300
fclock = pygame.time.Clock()

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)
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = - speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        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(伸缩型)吧

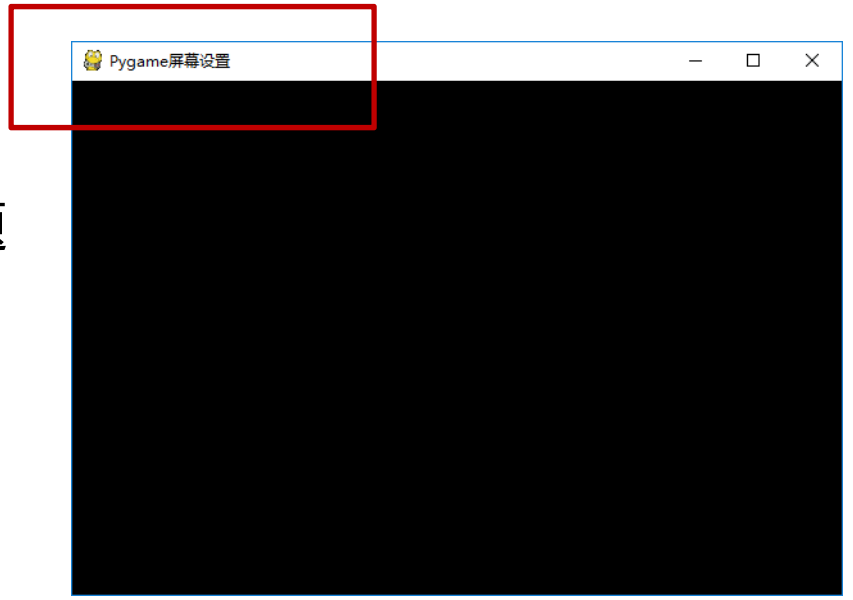


Pygame窗口标题和图标设置

窗口标题和图标函数

```
pygame.display.set_caption(title, icontitle=None)
```

- title设置窗口的标题内容
- icontitle设置图表化后的小标题
  - 小标题可选，部分系统没有



窗口标题和图标函数

```
pygame.display.get_caption()
```

- 返回当前设置窗口的标题及小标题内容
- 返回结构为(title, icontitle)
- 该函数与游戏交互逻辑配合，可以根据游戏情节修改标题内容

窗口标题和图标函数

```
pygame.display.set_icon(surface)
```

- 设置窗口的图标效果
- 图标是一个Surface对象

https://python123.io/PY15/PYG03-flower.png

https://python123.io/PY15/PYG03-withered-flower.png

PYG03-fire-flower.png **PYG03-ice-flower.png**

PYG03-pink-flower.png **PYG03-blue-flower.png**

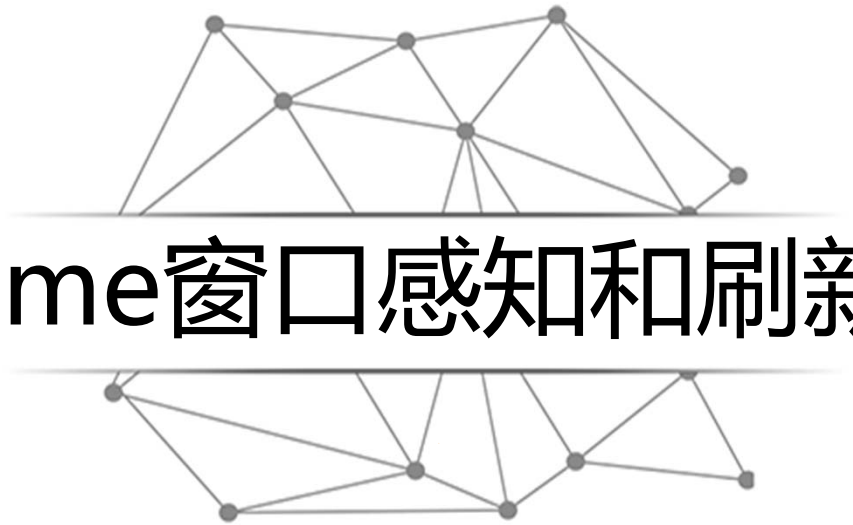
窗口标题和图标函数

pygame.display.set_icon(surface)

```
# Unit PYG03: Pygame Display Set
import pygame,sys

pygame.init()
icon = pygame.image.load("PYG03-flower.png")
pygame.display.set_icon(icon)
screen = pygame.display.set_mode((600,400), pygame.RESIZABLE)
pygame.display.set_caption("Pygame屏幕设置")

while True:
.....
```

```
# Unit PYG03: Pygame Wall Ball Game version 5  图标版
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()

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)
    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = - speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        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(图标型)吧



Pygame窗口感知和刷新运用

屏幕控制的重要函数

`pygame.display.get_active()`

- 当窗口在系统中显示(屏幕绘制/非图标化)时返回True，否则返回False

- 该函数可以用来判断是否游戏窗口被最小化
- 进一步，判断后可以暂停游戏，改变响应模式等

屏幕控制的重要函数

pygame.display.flip()

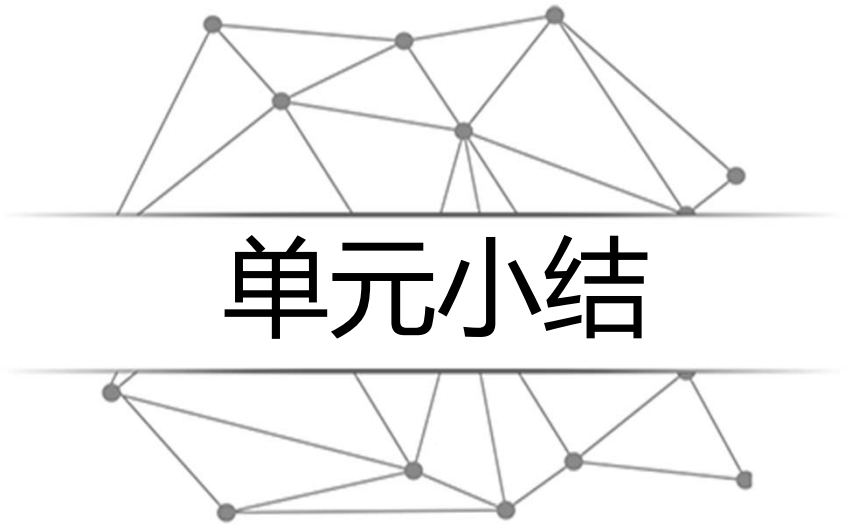
- 重新绘制整个窗口

pygame.display.update()

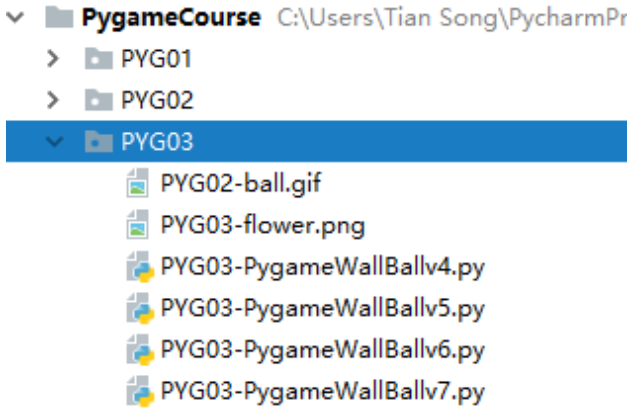
- 仅重新绘制窗口中有变化的区域，相比.flip()执行更快

与老师一起改改壁球小游戏(感知型)吧

```
# Unit PYG03: Pygame Wall Ball Game version 7  感知版
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()

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)
    if pygame.display.get_active():
        ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right > width:
        speed[0] = - speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = - speed[1]

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



单元小结



- Pygame屏幕绘制机制简介
- Pygame屏幕尺寸和模式设置
- Pygame窗口标题和图标设置
- Pygame窗口感知和刷新运用

Pygame屏幕绘制机制