



Pygame游戏设计

第04章 让游戏更酷一点

Python编程课

课程内容

- 游戏全屏
- 用图片当背景
- *改变小飞机大小



set_mod()

- 前面我们写的程序，都是通过pygame.display.set_mode()方法来指定游戏窗口的大小，并且返回一个surface对象，首先，我们先详细讲解一下set_mode()方法。

set_mode(resolution=(0, 0), flags=0, depth=0) -> surface

- resolution是一个二元组，表示宽和高，用于指定游戏界面的大小。
- flags用来指定附加选项，指定你想要显示的类型，多个附加选项用管道操作符'|'添加，比如传入FULLSCREEN | HWSURFACE。
- depth表示颜色深度，一般不设置。

选项	含义
FULLSCREEN	全屏模式
DOUBLEBUF	双缓冲模式
HWSURFACE	硬件加速支持（只有在全屏模式下才能使用）
OPENGL	使用OpenGL渲染
RESIZABLE	使得窗口可以调整大小
NOFRAME	使得窗口没有边框和控制按钮



全屏玩游戏

- 增加全屏的功能（按F11全屏，再按F11退出全屏）

```
.....[省略代码]
pygame.init()
bg = (0, 255, 0)
speed = [0, 0]
fullscreen = False #设置游戏窗口全屏的标志, True表示全屏, False表示窗口
.....[省略代码]
while True:
    for event in pygame.event.get():
        .....[省略代码]
        if event.type == KEYDOWN:
            if event.key == K_LEFT:
                speed = [-1, 0]
            .....[省略代码]
            if event.key == K_F11: #按下F11按键是切换全屏及窗口
                fullscreen = not fullscreen #全屏标志切换
                if fullscreen:
                    #第一个参数为电脑全屏的分辨率大小, 第二个参数为全屏且启用硬件
                    screen = pygame.display.set_mode((1366, 768), FULLSCREEN |
HWSURFACE)
                else:
                    screen = pygame.display.set_mode(screen_size) #切换为窗口
            else:
                speed = [0, 0]
```

加速



获取电脑分辨率

- 但是有一个问题，如果我的电脑分辨率不是1366*768，怎么办？
- 我们通过下面的程序获取自己电脑支持的分辨率

```
import pygame

pygame.init()
#获取电脑支持的分辨率列表
modes_list = pygame.display.list_modes()
print(modes_list)
```

- 得到的结果如下，全屏的分辨率为列表中的第一个元素：
[(1366, 768), (1360, 768), (1280, 768), (1280, 720), (1280, 600), (1024, 768), (800, 600), (640, 480), (640, 400), (512, 384), (400, 300), (320, 240), (320, 200)]



获取电脑最大的分辨率

- 自动获取电脑的全屏分辨率，使得游戏适应不同的电脑。

```
# 初始化pygame
pygame.init()

# 设置窗口大小
screen_size = width, height = (480, 650)
# 生成窗口对象
screen = pygame.display.set_mode(screen_size)
# 设置全屏开关
fullscreen = False
# 获取电脑屏幕所有的分辨率
modes_list = pygame.display.list_modes()
# 提取出全屏的分辨率
screen_full = modes_list[0]
```



```
.....[省略代码]
while True:
    for event in pygame.event.get():
        .....[省略代码]
        if event.type == KEYDOWN:
            if event.key == K_LEFT:
                speed = [-1, 0]
                .....[省略代码]
                #按下F11按键是切换全屏及窗口
                if event.key == K_F11:
                    fullscreen = not fullscreen #全屏标志切换
                    if fullscreen:
                        screen = pygame.display.set_mode(screen_full, FULLSCREEN |
HWSURFACE)
                    else:
                        #切换为窗口模式
                        screen = pygame.display.set_mode(screen_size)
            else:
                speed = [0, 0]
```



飞机飞行范围

- 然而，还有一个问题，为什么全屏以后，小飞机能移动的区域还是只有原来窗口的大小呢？
- 因为全屏以后游戏界面的宽和高都变大了，而我们之前的程序中小飞机飞出去的限制条件仍为原来小窗口的宽和高。所以窗口需要进行调整。



飞机飞行范围

- #按下F11按键是切换全屏及窗口

```
if event.key == K_F11:
```

```
    fullscreen = not fullscreen
```

```
    if fullscreen:
```

```
        # 设置成全屏
```

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

```
        # 将窗口的长和宽设置为全屏的数据
```

```
        width, height = screen_full
```

```
    else:
```

```
        # 设置成原始大小
```

```
        screen = pygame.display.set_mode(screen_size)
```

```
        # 将窗口的长和宽设定为原始数据
```

```
        width, height = screen_size
```

完成!



用图片当背景

- 游戏背景太单调了，我们来加个背景图吧。

```
# 设置标题
pygame.display.set_caption( "飞机大战" )
# 设置背景，读取一张图片，用作窗口模式
bg = pygame.image.load("data/images/background.png")
```

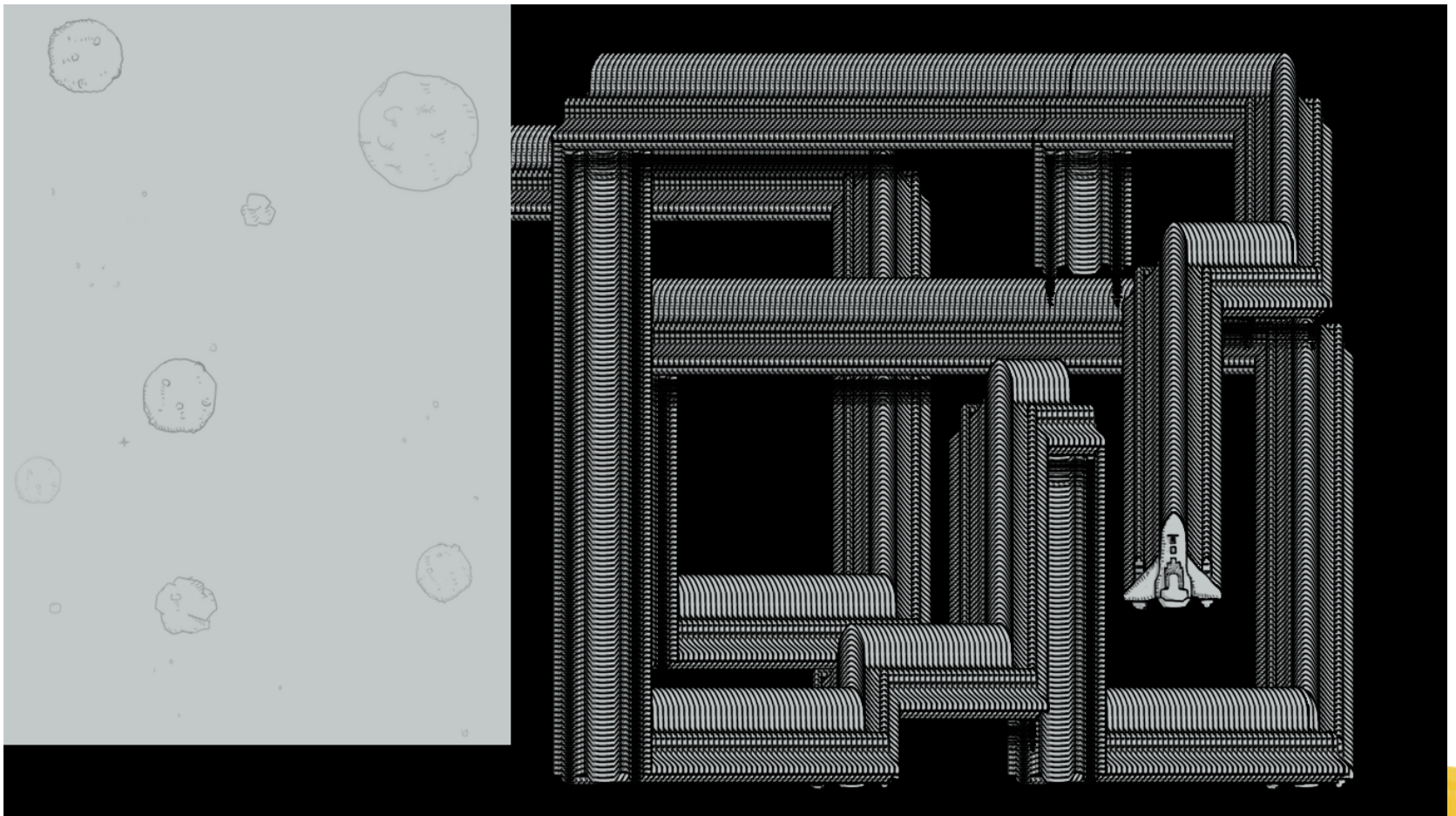
....[省略代码]

```
# 修改主循环中的背景填充方式(重要)
screen.blit(bg, (0, 0))
```



飞机穿越了？

但是游戏全屏的时候，背景图片没有全屏，飞机好像跑到了五维空间？



transform模块操作图像

- **transform**模块是pygame中用来改变或变换surface对象（图像）的模块。
- 比如我们第二节课用到的pygame.transform.flip()方法是用来水平或者垂直翻转图像的。
- pygame.transform.scale()方法可以按照一定的尺寸缩放图像。

方法	作用
flip	上下、左右翻转图像
scale	缩放图像（快速）
rotate	旋转图像
rotozoom	缩放并旋转图像
scale2x	快速放大一倍图像
smoothscale	平滑缩放图像（精准）
chop	裁剪图像



缩放背景图片

- 全屏和退出全屏时，对游戏背景图片的缩放操作。

```
# 设置标题
pygame.display.set_caption( "飞机大战" )

# 设置背景，读取一张图片，用作窗口模式
bg= pygame.image.load( "data/images/background.png" )
# 将背景图片进行缩放，大小和屏幕大小一致
```



- #按下F11按键是切换全屏及窗口
if event.key == K_F11:
 fullscreen = not fullscreen
 if fullscreen:
 # 设置成全屏
 screen = pygame.display.set_mode(modes_list[0], FULLSCREEN
| HWSURFACE)
 # 将窗口的长和宽设置为全屏的数据
 width, height = screen_full
 # 将背景图片缩放至全屏的大小
 bg = pygame.transform.scale(bg, screen_full)
 else:
 # 设置成原始大小
 screen = pygame.display.set_mode(screen_size)
 # 将窗口的长和宽设定为原始数据
 width, height = screen_size
 # 将背景图片缩放至窗口的大小
 bg = pygame.transform.scale(bg, screen_size)



提升运行速度

- 有没有发现一个问题，添加背景图以后小飞机移动的慢了，没有纯色背景的时候移动的快。下面我们优化一下程序，使得程序效率更高，小飞机移动的会快一点。
- #利用surface对象的convert_alpha()方法或convert()方法，加载图片并转换图片的像素格式
- convert()用于普通图片，没有透明部分，比如这里的背景图片

```
airPlane = pygame.image.load("me1.png").convert_alpha()
```

- convert_alpha()用于有透明部分的图片，比如这里的飞机图片

```
obg = pygame.image.load("background.png").convert()
```

试一试！
是不是快了许多？



*变换飞机大小

- 我们在玩一些其他游戏的时候，会遇到游戏人物变大变小的效果，比如超级玛丽、球球大作战。下面我们要
- 让我们的小飞机能够变大变小，通过按键控制变大变小。
- 我们使用smoothscale()方法，平滑缩放图像，而且更加精确。



- 程序实现，首先定义缩放的比率，加载小飞机的图像。
- `ratio = 1.0` #定义小飞机缩放的比率，初始化为1.0，即原始比例

#加载小飞机图像，缩放的时候以airPlane为原始图

`airPlane = pygame.image.load("me1.png")`

`newAirplane = airPlane` #定义newAirplane，用于保存缩放后的图像

`airPlane_rect = airPlane.get_rect()` #获取小飞机图像的rect对象

#position是用于更新小飞机位置的rect对象，newAirplane_rect用来保存缩放后小飞机图像的rect对象

`position = newAirplane_rect = airPlane_rect`



- 按 '=' 按键 (K_EQUALS) 放大, 按 '-' 按键 (K_MINUS) 缩小, 按空格按键 (K_SPACE) 返回原始尺寸。
- `if event.type == KEYDOWN:`
[省略代码]
`if event.key == K_EQUALS or event.key == K_MINUS or event.key == K_SPACE:`
 `#最大放大比例不能超过2倍`
 `if event.key == K_EQUALS and ratio < 2.0:`
 `ratio += 0.1 #每次按下 '=' 按键, 放大0.1倍`
 `#最小缩小比例不能低于0.5倍`
 `if event.key == K_MINUS and ratio > 0.5:`
 `ratio -= 0.1 #每次按下 '-' 按键, 缩小0.1倍`
 `if event.key == K_SPACE:`
 `ratio = 1.0 #按下空格键, 返回原始比例`
 `#在原始图片airPlane基础上按一定的大小 (width*height) 缩放, 并返回新的`
 `#surface对象给newAirplane, 缩放的大小尺寸必须用整数, 所以要强制转换成int型`
 `newAirplane = pygame.transform.smoothscale(airPlane, \`
 `(int(airPlane_rect.width * ratio), \`
 `int(airPlane_rect.height * ratio)))`
 `newAirplane_rect = newAirplane.get_rect() #获取缩放后的小飞机的rect对象`
 `#将缩放后的小飞机的左上角位置赋值给position`
 `position.width, position.height = newAirplane_rect.width,`
 `newAirplane_rect.height`



- 绘制缩放后的小飞机以及整个画面。
- ```
screen.blit(bg, (0, 0))
screen.blit(newAirplane, position) #这里要用缩放后新的小飞机图像
newAirplane绘制
pygame.display.flip()
```

按下缩放按键，  
试一试吧！

