

# alien\_run 设计文档

---

## 游戏说明

对flipped bird项目进行修改,增加了速度控制,背景音乐控制,速度提示,随机难度等功能

## 玩法介绍

键盘控制:

↑: 精灵向上运动

↓: 精灵加速向下运动

←: 精灵向上运动,同时增加运动速度

→: 精灵向上运动,同时增加运动速度

SPACE: 背景音乐暂停与播放

## 参考资料

[Welcome to Pygame Zero](#)

[Games with PyGame Zero](#)

## 关键处理流程

完整代码及素材已经放在github上了

[github链接](#)

- 障碍物的定义和移动

```
#初始化障碍物
def reset_pipes():
    GAP = random.randint(250,350)
    pipe_gap_y = random.randint(200, HEIGHT - 200)
    pipe_top.pos = (WIDTH, pipe_gap_y - GAP // 2)
    pipe_bottom.pos = (WIDTH, pipe_gap_y + GAP // 2)

    pipe_top = Actor('top', anchor=('left', 'bottom'))
    pipe_bottom = Actor('bottom', anchor=('left', 'top'))
    reset_pipes() # Set initial pipe positions.

def update_pipes():
```

```

global SPEED
pipe_top.left -= SPEED
pipe_bottom.left -= SPEED

if pipe_top.right < 0:
    reset_pipes()
    if not alien.dead:
        alien.score += 1
        SPEED += 0.5
        //速度增加

```

- 精灵的移动

```

def update_alien():
    uy = alien.vy
    alien.vy += GRAVITY
    alien.y += (uy + alien.vy) / 2
    alien.x = 75

    if not alien.dead:
        alien.image = 'alien'

    if alien.colliderect(pipe_top) or alien.colliderect(pipe_bottom):
        alien.dead = True
        alien.image = 'alien_hurt'
        sounds.a.play()
        //死亡音效

    if not 0 < alien.y < 720:
        global SPEED
        alien.y = 200
        alien.dead = False
        alien.score = 0
        alien.vy = 0
        SPEED = 3
        reset_pipes()

```

- 键盘控制,上下左右空格

```

def on_key_down():
    global SPEED
    global pause
    if not alien.dead:
        if keyboard[keys.UP]: //上
            alien.vy = -FLAP_STRENGTH
            sounds.jiujiu.play()
        if keyboard[keys.LEFT]: //左
            alien.vy = -FLAP_STRENGTH
            sounds.jiujiu.play()
            SPEED -= 0.5
        if keyboard[keys.RIGHT]: //右

```

```

        alien.vy = -FLAP_STRENGTH
        sounds.jiujiu.play()
        SPEED +=0.5
    if keyboard[keys.DOWN]:          //下
        alien.vy = FLAP_STRENGTH
        sounds.jiujiu.play()
        SPEED +=0.5
    if keyboard[keys.SPACE]: //空格
        if pause:
            music.pause()           //背景音乐暂停
            pause = False
        else:
            music.unpause()         //背景音乐播放
            pause = True

```

- 背景图片,速度,成绩的打印,死亡提示

```

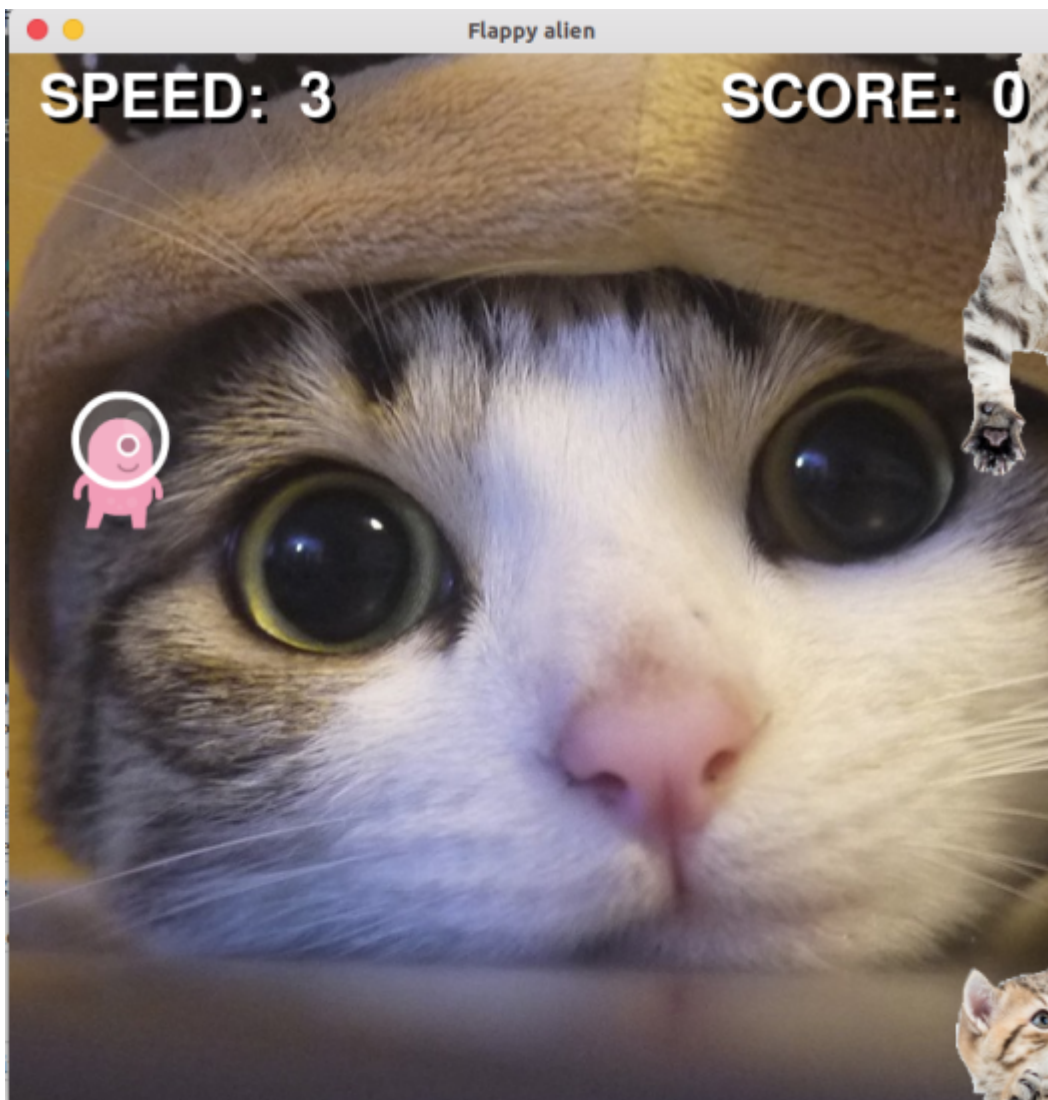
def draw():
    screen.blit('bg', (-550,-350))          //背景图片
    pipe_top.draw()
    pipe_bottom.draw()
    alien.draw()
    global SPEED
    global color1
    if SPEED>7:                               //速度字体颜色控制
        color1='YELLOW'                     //超过7的时候
    else:                                     //颜色自动变黄
        color1='white'
    if alien.score>7:                         //得分字体颜色控制
        color2='RED'                       //超过7的时候
    else:                                    //颜色自动变红
        color2='white'
    screen.draw.text(
        "SPEED: "+str(SPEED),              //显示速度
        color=str(color1),
        midtop=(120,10),
        fontsize=60,
        shadow=(1,1)
    )
    screen.draw.text(
        "SCORE: "+str(alien.score), //显示分数
        color=color2,
        midtop=(WIDTH-120, 10),
        fontsize=60,
        shadow=(1, 1)
    )
    if alien.dead:                           //死亡提醒
        screen.draw.text(
            "You died!",
            color='RED',
            midtop=(WIDTH//2, HEIGHT//2),

```

```
        fontsize=100,  
        shadow=(1, 1)  
    )
```

## 演示截图

开始界面



游戏界面



死亡界面

