第23课：背景音乐和背景图片

步骤目标

本文的第一个目标是实现播放背景音乐功能。游戏程序启动后，背景音乐开始播放。玩家按下 m 字母键，暂停播放音乐；再次按下 m 字母键，继续播放音乐。

第二个目标是在游戏窗口中添加背景图片。完成后的效果如图1所示。

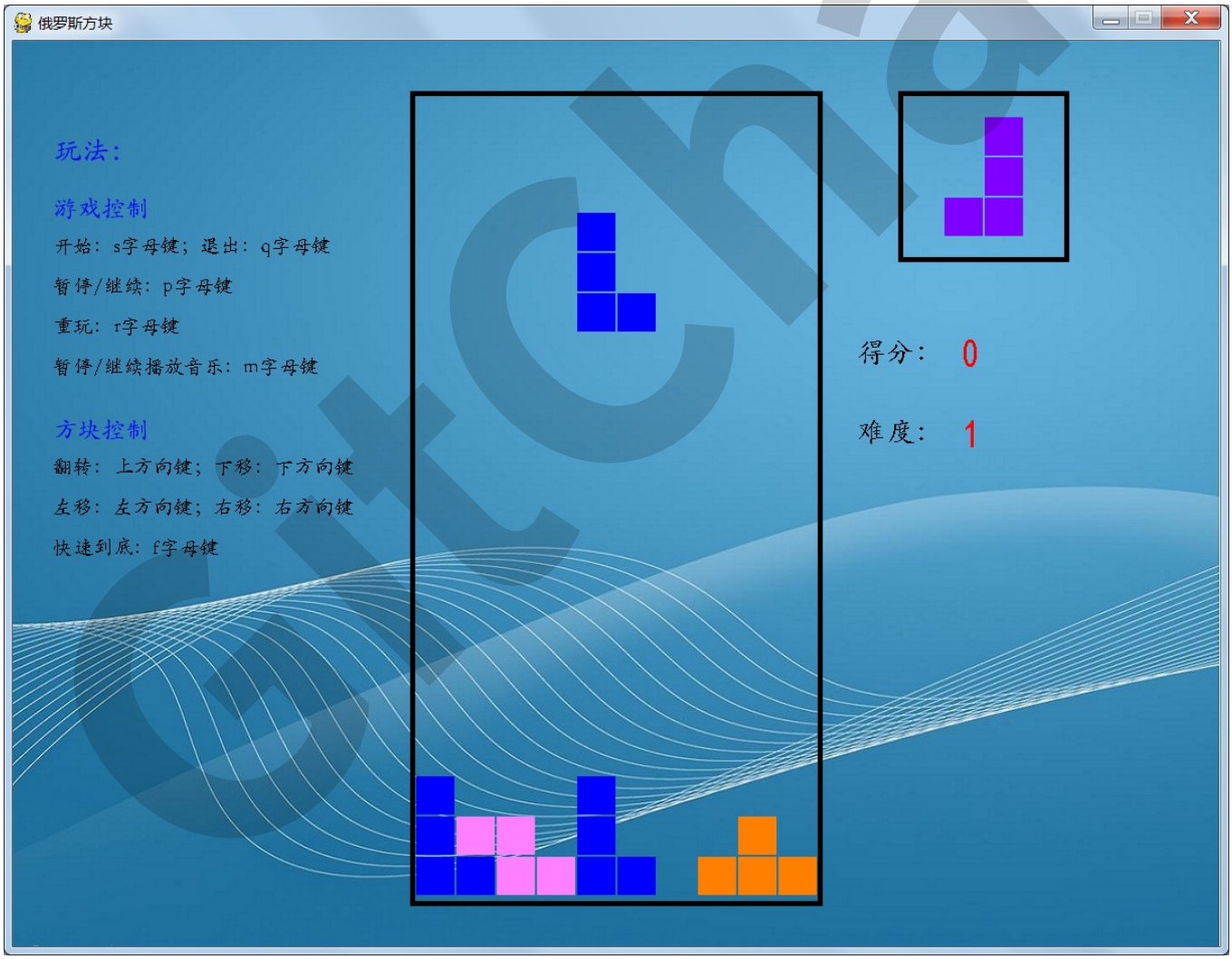

图1 窗口添加背景图片

本步骤的任务是：

1. 加载 MP3 音乐文件并开始播放音乐。
2. 暂停/继续播放音乐。
3. 添加背景图片。

加载和播放音乐文件

Pygame 包的 mixer 模块可以加载和播放音乐。我们在 GameResource 类内定义 `play_bg_music()` 方法来完成加载和播放音乐功能，代码如下。

```
1.  def play_bg_music(self):
2.          pygame.mixer.init()
3.          bg_music_file = self.music_path + 'bg_music.mp3'
4.          pygame.mixer.music.load(bg_music_file)
5.
6.          pygame.mixer.music.play(-1)    #-1是指循环播放
```

要使用 pygame.mixer，首先要调用 pygame.mixer.init() 进行初始化。然后，可以调用 pygame.mixer.music.load() 来加载音乐文件，参数就是音乐文件的路径。接着调用 pygame.mixer.music.play() 来播放音乐。

我们准备了 `bg_music.mp3` 音乐文件，放在 music 文件夹内，如图2所示。GameResource 的 music_path 属性值为 `music/`。

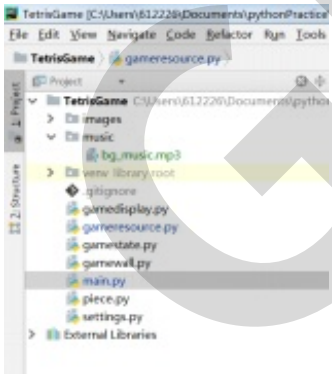

图2 music 文件夹内放置音乐文件

在哪里调用 `play_bg_music()` 方法呢？我们是在 main.py 的 main 函数内调用它。这使得程序启动后就会播放背景音乐。

```
def main():
    #初始化pygame。启用Pygame必不可少的一步，在程序开始阶段执行。
    pygame.init()
        ......    #部分代码省略

    game_state = GameState(screen)
    game_resource = GameResource()
    game_resource.play_bg_music()
#游戏主循环
    while True:
        ......    #省略部分代码
```

需要指出的是，必须在调用 pygame.init() 之后调用 pygame.mixer.init() 方法。

暂停和继续播放音乐

玩家按下 m 字母键后，暂停播放音乐。此时，我们把 GameResource 类实例的 `music_paused` 属性值设为 True。玩家再次按下 m 字母键，继续播放音乐。此时，把 `music_paused` 属性值设为 False。我们用 `pause_bg_music()` 方法来封装暂停和继续播放音乐功能。

```
def pause_bg_music(self):
    if not self.music_paused:
        pygame.mixer.music.pause()
        self.music_paused = True
    else:
        pygame.mixer.music.unpause()
        self.music_paused = False
```

调用 pygame.mixer.music.pause() 能够暂停播放。调用 pygame.mixer.music.unpause() 则取消暂停，继续播放音乐。

显然，我们要在 main.py 文件的 `on_key_down` 函数内添加响应 m 字母键按下事件的代码。哪些地方做了修改，请看注释说明。

```

def check_events(game_state, game_resource):  #此处有修改
    '''捕捉和处理键盘按键事件'''
```

```
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                sys.exit()
            elif event.type == pygame.KEYDOWN:
                on_key_down(event, game_state, game_resource) #此处有修改
            elif event.type == pygame.USEREVENT:
                game_state.piece.move_down()


    def on_key_down(event, game_state, game_resource): #此处有修改
        ......  #与前一版本相同，故省略
        elif event.key == pygame.K_m:  #此处有修改
            game_resource.pause_bg_music() #此处有修改
```

以上改动包括：

1. `check_events` 函数和 `on_key_down` 函数内添加了 `game_resource` 参数。相应地，main 函数内调用 `check_events` 函数的时候要传入 `game_resource`，即：

```
    #监视键盘和鼠标事件
    check_events(game_state, game_resource)
```

2. 在 `on_key_down` 函数内添加响应 m 字母键按下事件的代码，也即上面最后两行代码。

留点小作业给你，请你实现玩家按下 q 键退出程序的功能。

绘制背景图片

做法上，绘制背景图片与以前的第15或第16课介绍的绘制图片是一样的。做法是：

1. 在 images 子文件夹内添加图片文件。背景图片的文件名是 bg-1.jpg。图片文件可从文末给出链接下载，或者你自己制作一份（分辨率是1200*900）。
2. 在 GameResource 类内定义 `bg_img` 属性和 `load_bg_img()` 方法。`load_bg_img` 方法的定义如下。

```
def load_bg_img(self):
    if not self.bg_img:
        self.bg_img = pygame.image.load(self.img_path + "bg-1.jpg")
```

```
4.
5.         return self.bg_img
```

3. 在 main.py 文件的 main 函数内，进一步是在主循环内，设置屏幕背景色，代码如下：

```
1.  #设定屏幕背景色
2.  screen.fill(bg_color)
```

改为设置背景的语句：

```
1.  #设定屏幕背景
2.  screen.blit(game_resource.load_bg_img(), (0, 0))
```

函数 blit 的第一个参数是图片资源对象。调用 `game_resource.load_bg_img()` 将返回图片资源对象。第二个参数(0, 0)是图片显示位置，即从窗口左上角开始绘制图片。

注意，绘制背景图片的动作必须放在绘制其他窗口元素的动作之前，否则后者会被“遮挡”。

小结

本步骤实现了以下功能：

1.背景音乐。使用 pygame.mixer 模块来加载和播放背景音乐。玩家按下 m 字母键，将暂停播放背景音乐；再次按下 m 字母键，将继续播放背景音乐。
2.背景图片。准备好一张与窗口分辨率一致的图片，从窗口左上角开始绘制图片即可。

实现背景音乐和背景图片功能的代码可从以下链接浏览或下载。

- Github

这一版本还包含了一个小功能，就是玩家按 q 字母键后，程序立即退出。

至此，我们实现了具有基本功能的俄罗斯方块游戏，整个开发过程就结束了。如果你有兴致，你可以扩充功能。比如：

1. 游戏刚开始的时候，游戏区域就出现一堵墙，要求玩家在有墙的前提下玩游戏。
2. 消行的时候，发出有赞赏效果的音效。

3. 两个玩家同时参与，进行竞赛。

如果你对前面实验步骤还不够熟悉，我建议你再重新做一遍。有想法有追求的程序员会这么做，因为再来一遍会学得更多，更扎实。