

第02课：显示游戏窗口

步骤目标

本文将达成的目标是：开发完成一个 Python 程序，运行该程序将显示图1所示的俄罗斯方块游戏窗口。



图1 俄罗斯方块游戏窗口

完成本目标的前提是，你在自己的电脑中安装了 Python 3 和 Pycharm（用免费的社区版即可），而且你已经针对俄罗斯方块游戏开发创建了工程，并安装了 Pygame 包。如果你没有做到以上前提，请返回到第一篇，那里讲了怎么做。

显示游戏窗口功能实现的步骤分成两个子步骤：

1. 创建和编写程序文件 main.py。
2. 运行 main.py。

显示游戏窗口功能实现

上一篇中，我们已经创建了 TetrisGame 工程（工程名字可以换成别的）。下面让我们在该工程下创建第一个 Python 程序文件 main.py。你可以把 main.py 换成其他文件名。我之所以使用 main.py 这个名字，是因为它表明该文件是程序运行开始的地方。

创建和编写程序文件 main.py

（1）创建程序文件 main.py。

图2演示了如何在 TetrisGame 工程内创建 Python 程序文件。做法是鼠标右键点击工程 TetrisGame，然后在弹出的菜单中选择“New/Python File”，在弹出的小窗口内输入程序文件名：main，点击“OK”按钮后将创建 main.py 文件。

要强调的是，New 菜单的下级菜单中要选择的是处于中部的“Python File”，不是顶部的“File”。

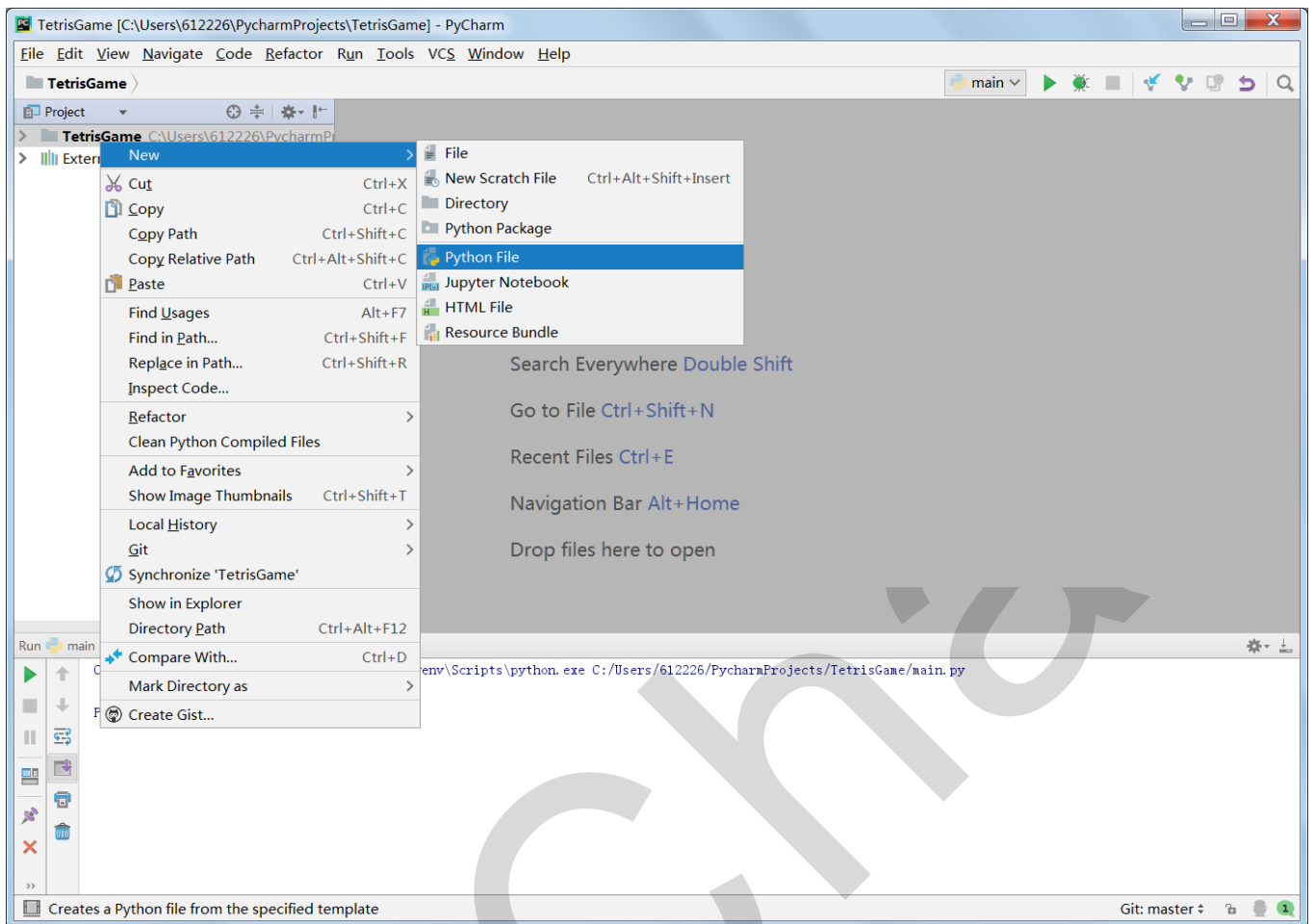


图2 创建程序文件

(2) 编写main.py的代码。

将下文展示的代码（最左侧的数字序号是代码行号），敲入 main.py 程序文件内。前面四行是注释，不需要敲进程序文件内。

运行 main.py

鼠标右键选中“main.py”，在弹出的菜单中点击“Run ‘Main’”菜单项运行程序。程序运行后，将弹出游戏窗口，见图1。

下文将对 main.py 中的代码及程序的运行流程做详细解读。

程序文件 main.py 的说明

实现显示游戏窗口的代码 main.py，如下

```

1. 1. # @Time      : 2018/4/23 16:44
2. 2. # @Author   : freedomyeah
3. 3. # @Email    : iamdouble@163.com
4. 4. # @Copyright: MIT
5. 5. import sys
6. 6. import pygame
7. 7.
8. 8. def main():
9. 9.     #初始化pygame。启用Pygame必不可少的一步，在程序开始阶段执行。
10. 10.    pygame.init()
11. 11.    #创建屏幕对象（也即窗口对象）
12. 12.    screen = pygame.display.set_mode((1200, 900)) #分辨率是1200*900
13. 13.    pygame.display.set_caption("俄罗斯方块") #窗口标题
14. 14.
15. 15.    #屏幕背景色
16. 16.    bg_color = (230, 230, 230)
17. 17.
18. 18.    #游戏主循环
19. 19.    while True:
20. 20.        #监视键盘和鼠标事件
21. 21.        for event in pygame.event.get():
22. 22.            if event.type == pygame.QUIT: #关闭窗口的事件
23. 23.                sys.exit() #退出程序
24. 24.
25. 25.        #填充屏幕背景色
26. 26.        screen.fill(bg_color)
27. 27.
28. 28.        #刷新屏幕
29. 29.        pygame.display.flip()
30. 30.
31. 31. if __name__ == '__main__':
32. 32.    main()

```

对于上面的代码，说明如下：

- (1) 第5行导入 sys 模块。第23行调用了 sys 模块的 exit() 函数，作用是退出程序。
- (2) 第6行导入 pygame 模块。pygame 模块用于绘制图形和显示窗口。
- (3) 第8~29行定义了 main() 函数，然后在第32行调用 main()函数。第31行代码的作用是检测本行代码所在的程序文件（此处就是main.py）是否是“直接运行的情形”，如果是就调

用 main() 函数；否则不调用 main() 函数。

- 在 Pycharm 中右键点击 main.py 可直接运行它。此外，在命令行界面上，比如 Windows 终端上，输入 “python main.py”，也可直接运行 main.py。
- 什么时候 main.py 不属于直接运行的情形呢？答案是，别的程序文件（比如 abc.py）导入 main.py（即：import main.py）的时候。

（4）对于 main() 函数内的代码，解释说明如下。

- 第10行是调用 pygame 模块的 init 函数进行初始化。使用 pygame 函数之前务必进行初始化。
- 第12行是调用 display 模块的 `set_mode` 函数，作用是初始化屏幕对象（也即窗口对象）。此处传入一个参数，即(1200, 900)元组，这使得窗口的分辨率是1200x900。display 模块隶属于 pygame 模块。
- 第13行代码的作用是设置窗口标题。
- 第16行代码的作用是把颜色值(230, 230, 230)赋值给 `bg_color` 变量。第26行代码将把 `bg_color` 用作窗口背景色，效果如图1所示。颜色值(230, 230, 230)中，三个整数依次是三原色中红色、绿色和蓝色的浓度值。浓度值是一个整数，最大为255，最小为0。
- 第19~29行代码在下面解释说明。

（5）第19行~29行构成一个 while 循环。

- 第19行语句中，循环测试条件是 True，意味着这是一个无穷循环。也就是说，循环会一直执行，直至玩家关闭窗口。
- 第21行中，`pygame.event.get()` 的作用是获取事件列表。事件列表内包含0个或多个事件对象。第21行将依次赋值给 event 变量。

知识点： 玩家每次按键盘，将引发键盘按键事件，接着 pygame 将新建事件对象，然后事件对象将加入到事件列表中。玩家操作鼠标也会引发事件。

- 第22行代码中，event.type 是事件对象的类型。pygame.QUIT 是 pygame 模块内定义的常量，指的是关闭窗口的事件类型。玩家点击窗口标题栏上的关闭按钮（见图1右上角），将引发 pygame.QUIT 类型的事件。而玩家按下键盘的话，将引发 pygame.KEYDOWN 类型的事件。
- 第23行代码是在玩家点击窗口标题栏上的关闭按钮时执行的，作用是退出整个程序。
- 第26行是填充屏幕背景色。第29行是刷新屏幕，以使最近的绘制操作生效。

程序运行流程

- 遍历事件列表，对每一个事件作出响应处理。响应处理的结构大致是：

```
1.     if (是关闭窗口事件)
2.         退出程序
3.     elif (是键盘按键事件)
4.         处理按键事件
```

- 绘制窗口内的元素。例如填充背景色。以后会有更多绘制操作。
- 刷新窗口。

实现新功能的绝大多数代码，将插入游戏主循环内。当然，我们会把扩充的代码封装成一个个函数。

小结

正如开头所说，本文的目标是显示分辨率为1200x900的游戏窗口。如果你达成目标，说明你干得不错。

在理解各行代码用途的前提下，你可以抄代码，不必要记住它们。然而，你务必吃透程序主循环，以免将来对不断增加的代码失去掌控。

我已经把源码上传至 [GitHub](#)

你点击网页中的“main.py”（如图3所示），就可以浏览该文件。要下载整个工程文件夹（目前其下只有main.py文件），则先点击“Clone or download”，再点击“Download Zip”，你将下载到一份zip文件，解压后正是工程文件夹。

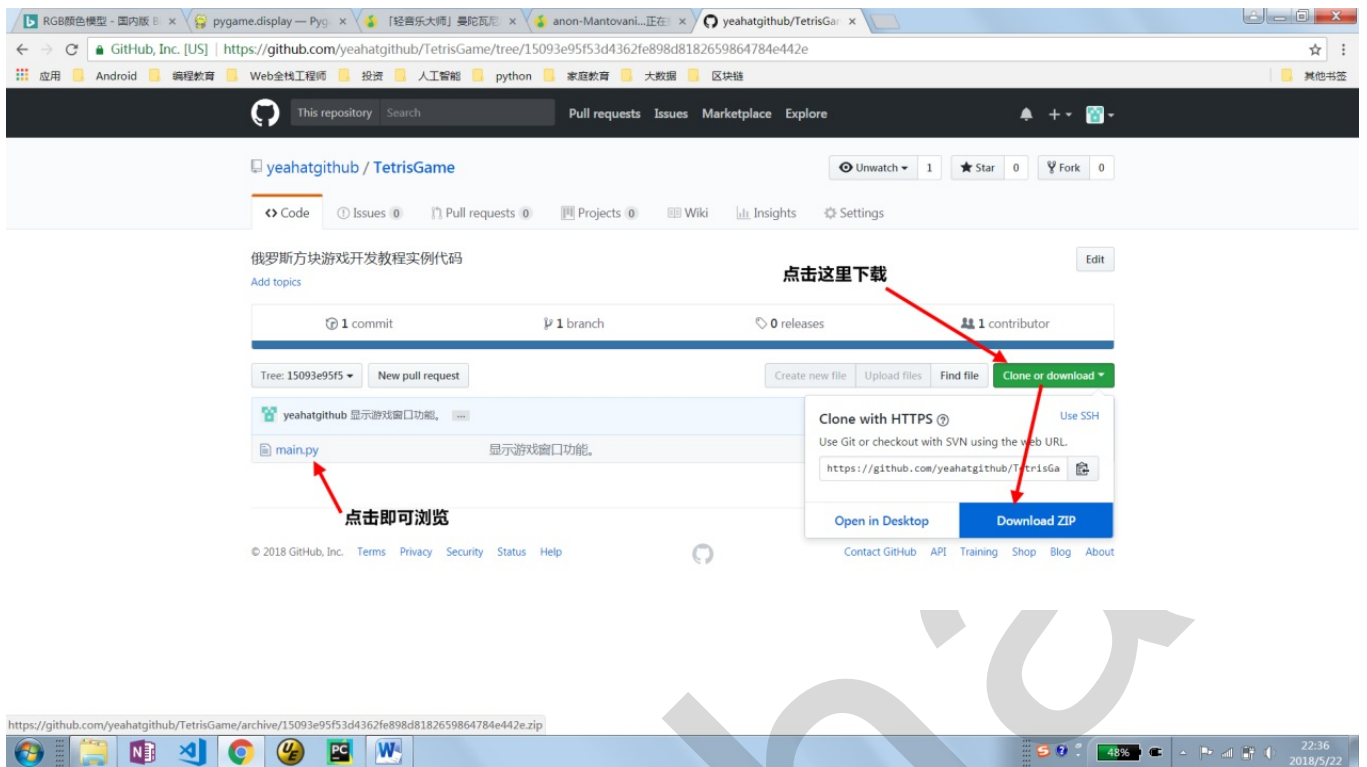


图3 在GitHub上浏览或下载源码