

第04课：绘制小方块

步骤目标

本文实现的目标是：在游戏区的左上角绘制一个小方块，如图1所示。

我们知道，俄罗斯方块是由四个小方块组合而成的。绘制方块等于绘制4个小方块。因此让我们从绘制小方块开始。

绘制小方块功能怎么实现的？实质是，绘制一个矩形。

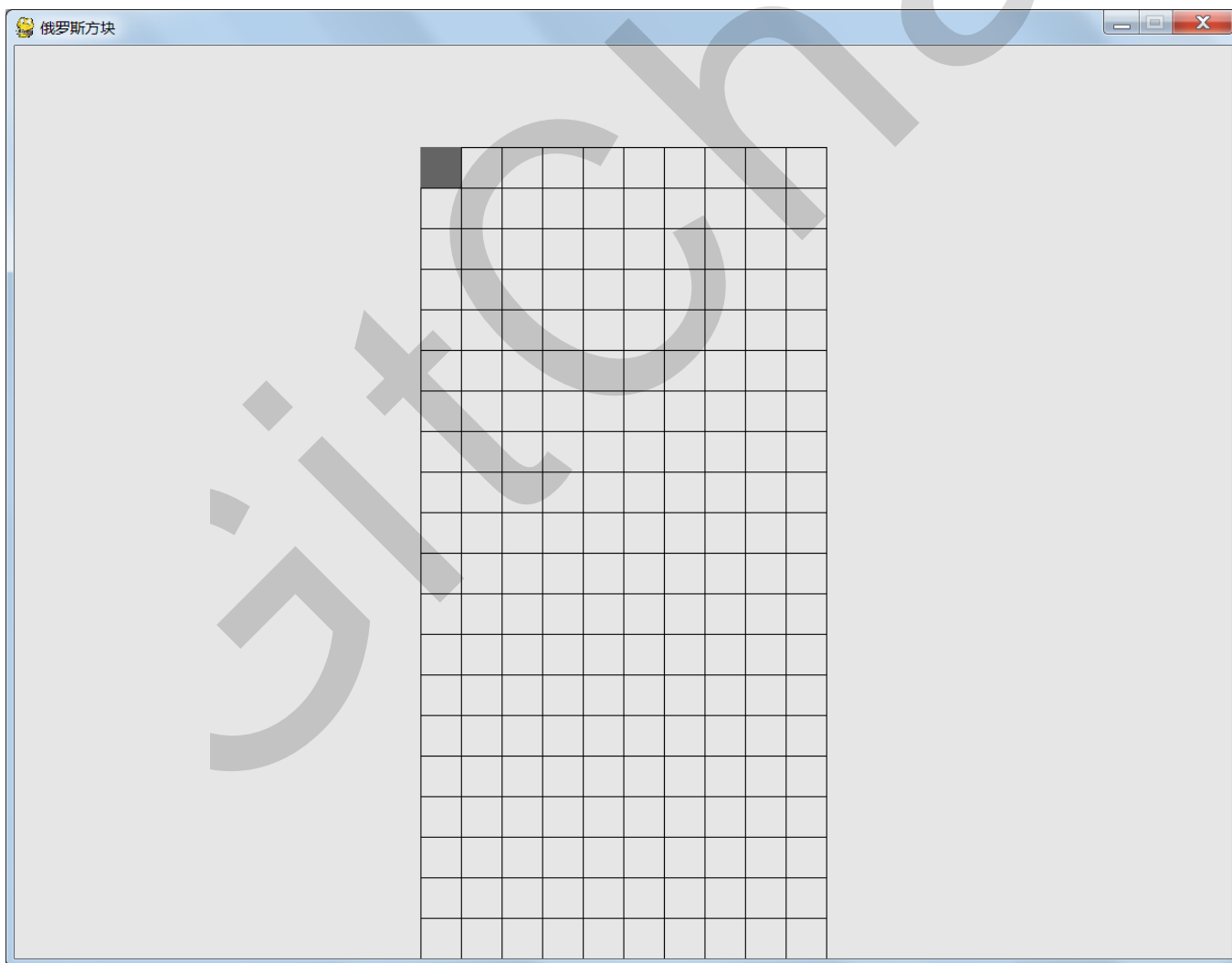


图1 绘制小方块

绘制小方块

实现绘制小方块的功能，与实现绘制矩形区域网格线的功能是类似的，实现代码如下所示。主要步骤有两个：

1. 在主循环中调用 `draw_cell()` 函数。见下面 TetrisGame/main.py 的第41，42行代码。
2. 定义 `draw_cell` 函数，函数代码完成了绘制小方块的功能。见 TetrisGame/main.py 第55~64行代码。

TetrisGame/main.py 中绘制小方块的代码（左侧的数字是代码行号），如下：

```
1.  # TetrisGame/main.py
2.  8   SCREEN_WIDTH = 1200           #窗口宽度
3.  9   SCREEN_HEIGHT = 900          #窗口高度
4.  10  CELL_WIDTH = 40               #方块在20*10个单元格组成的游戏区内移动。每个单元格的边长是40个像素。
5.  11  GAME_AREA_WIDTH = CELL_WIDTH * 10      #一行10个单元格
6.  12  GAME_AREA_HEIGHT = CELL_WIDTH * 20     #一共20行
7.  13  GAME_AREA_LEFT = (SCREEN_WIDTH - GAME_AREA_WIDTH) // 2      #游戏区左侧的空白区的宽度
8.  14  GAME_AREA_TOP = SCREEN_HEIGHT - GAME_AREA_HEIGHT            #游戏区顶部的空白区的宽度
9.  15  EDGE_COLOR = (0, 0, 0)      #游戏区单元格边界线的颜色。今后，网格线会被去除。
10. 16  CELL_COLOR = (100, 100, 100) #单元格填充色。
11. 17  BG_COLOR = (230, 230, 230)  #窗口背景色
12. 18
13. 19
14. 20  def main():
15. 21      #初始化pygame。启用Pygame必不可少的一步，在程序开始阶段执行。
16. 22      pygame.init()
17. 23      #创建屏幕对象
18. 24      screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT)) #分辨率是900*900
19. 25      pygame.display.set_caption("俄罗斯方块") #窗口标题
20. 26
21. 27      #屏幕背景色
22. 28      bg_color = BG_COLOR
23. 29
24. 30      #游戏主循环
25. 31      while True:
26. 32          #监视键盘和鼠标事件
```

```

27. 33         for event in pygame.event.get():
28. 34             if event.type == pygame.QUIT:
29. 35                 sys.exit()
30. 36
31. 37         # 填充屏幕背景色
32. 38         screen.fill(bg_color)
33. 39         #绘制20*10网格
34. 40         draw_game_area(screen)
35. 41         #绘制小方块
36. 42         draw_cell(screen, GAME_AREA_LEFT, GAME_AREA_TOP)
37. 43         #让最近绘制的屏幕可见
38. 44         pygame.display.flip()
39. 45
40. ... .. #代码未做改动，省略。其内定义了draw_game_area函数。
41. 55 def draw_cell(screen, left, top):
42. 56     '''绘制单元格，也即绘制小方块'''
43. 57     '''
44. 58     left: 单元格离窗口左边界的距离。单位是像素。
45. 59     top: 单元格离窗口上边界的距离。
46. 60     '''
47. 61     cell_left_top = (left, top)        #小方块的左上角坐标点
48. 62     cell_width_height = (CELL_WIDTH, CELL_HEIGHT)    #小方块的宽度和高度
49. 63     cell_rect = pygame.Rect(cell_left_top, cell_width_height)    #生成由左上角坐标和宽度高度定义的矩形
50. 64     pygame.draw.rect(screen, CELL_COLOR, cell_rect)    #绘制正方形
51. 65
52. 66
53. 67
54. 68 if __name__ == '__main__':
55. 69     main()

```

下面是代码解读：

第61行代码定义了一个元组(left, top)，用作设置将要绘制的矩形的左上角坐标。

第62行定义了要绘制的矩形的宽度和高度。该行代码所用到的 `CELL_WIDTH` 常量在第10行进行了定义。这一常量是单元格的边长，为40个像素。

第63行代码调用 `pygame.Rect` 函数生成了矩形对象，`Rect` 的中文含义是矩形。

第64行是在屏幕 `screen` 上，用 `CELL_COLOR` 颜色填充 `cell_rect` 所描述的矩形。该行代码

所用到的 `CELL_COLOR` 常量在第16行进行了定义。这一常量用作单元格填充色，颜色值采用 RGB 三原色模型来表示。

完成绘制小方块功能的代码，大家可以访问[这里](#)

你应该先自己做尝试，不要只是“阅读而后复制”。自己多尝试，能更快掌握编程技能。这不是你正在追求的？

作业

居中绘制小方块

你能够把小方块绘制到游戏区顶部的中央吗？就像图2所示。你要做的就是调整绘制小方块这个矩形的坐标。

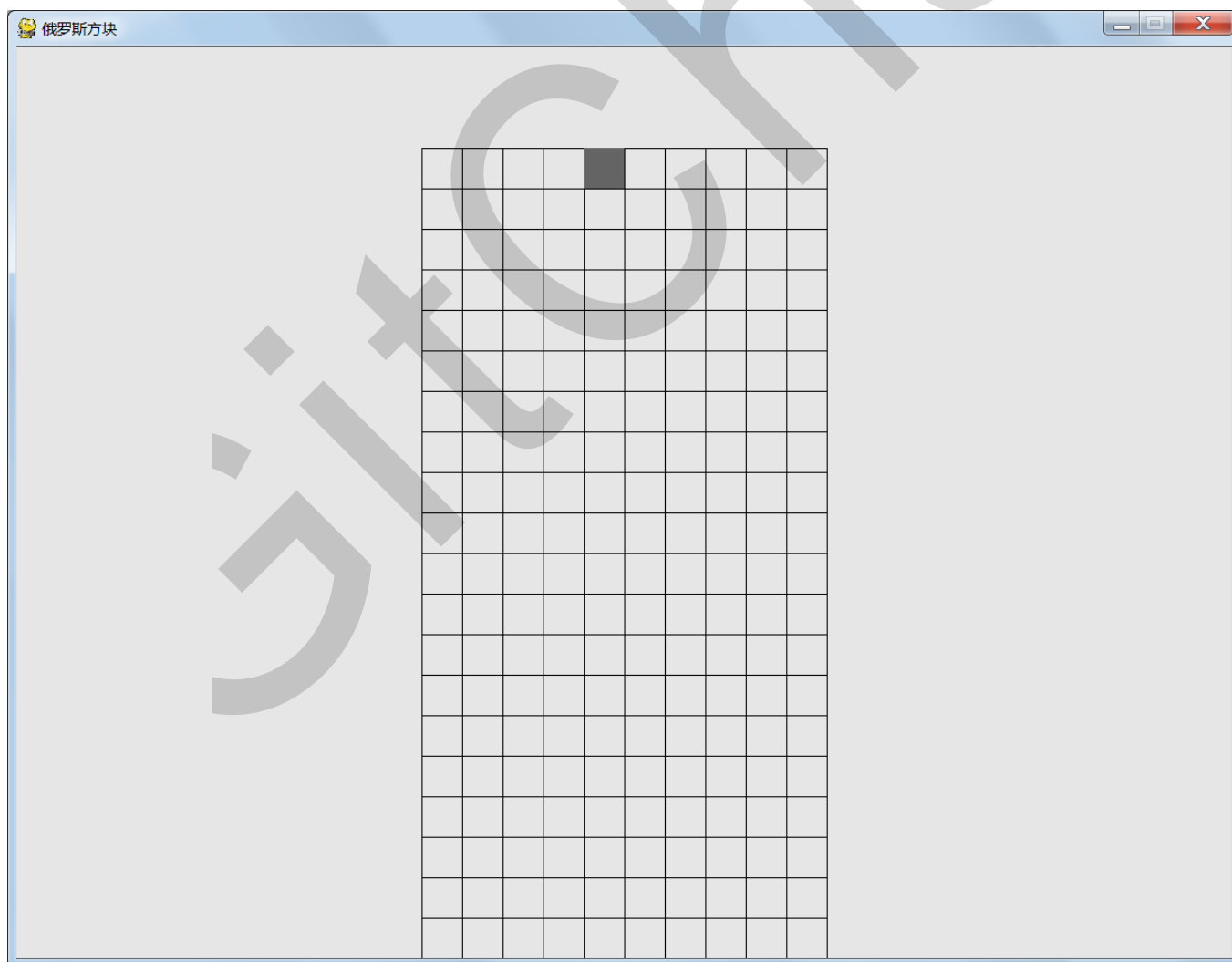


图2 把小方块绘制到游戏区顶部的中央

完成上述功能的代码，大家可以参照[这里](#)。

把常量定义移到 settings.py 文件内

上述 TetrisGame/main.py 代码中，第8行到17行都是定义常量的语句。请你尝试做两点：

1. 新建一个代码文件，取名为 settings.py。这一文件专门用来定义游戏程序用到的常量。
2. 把 main.py 内的常量定义语句移到 settings.py 文件。main.py 文件的第8行处要插入以下语句：

```
1.         from settings import *
```

这一语句的作用是从 settings 模块（即 settings.py 文件）导入所有的名字。这里就是所有的常量。

思考如何绘制方块

我们该如何在游戏区域内绘制方块呢？这一问题可以分解为以下几个子问题：

1. 如何记住方块的类型？
2. 如何根据方块的类型得出方块的形状？要知道，方块是由4个小方块（即单元格）拼成的。绘制方块就是绘制4个小方块。知道了方块类型，该绘制哪4个单元格呢？
3. 如何在游戏区域内定位方块？也就是说，要在游戏区域的哪个位置绘制方块？

下一篇文章将回答以上问题。这些问题的答案的要点在于用一个矩阵来表示方块的形状。比如，S型方块的矩阵是：

```
1.     .OO.  
2.     OO..  
3.     ....
```

上述表示中，O 字母表示被方块占据，点号表示给位置未被占据。

JitChia