

第21课：强制重新开始和界面变样

步骤目标

本文有两大目标。一是强制重新开始，即玩家按下 r 字母键，无论游戏是暂停中还是进行中，游戏都重新开始。估计，玩家认为这一局玩得太水了。二是，去掉游戏区域的网格线，调整游戏区域在窗口中的位置。这样做后，界面效果如图1所示。

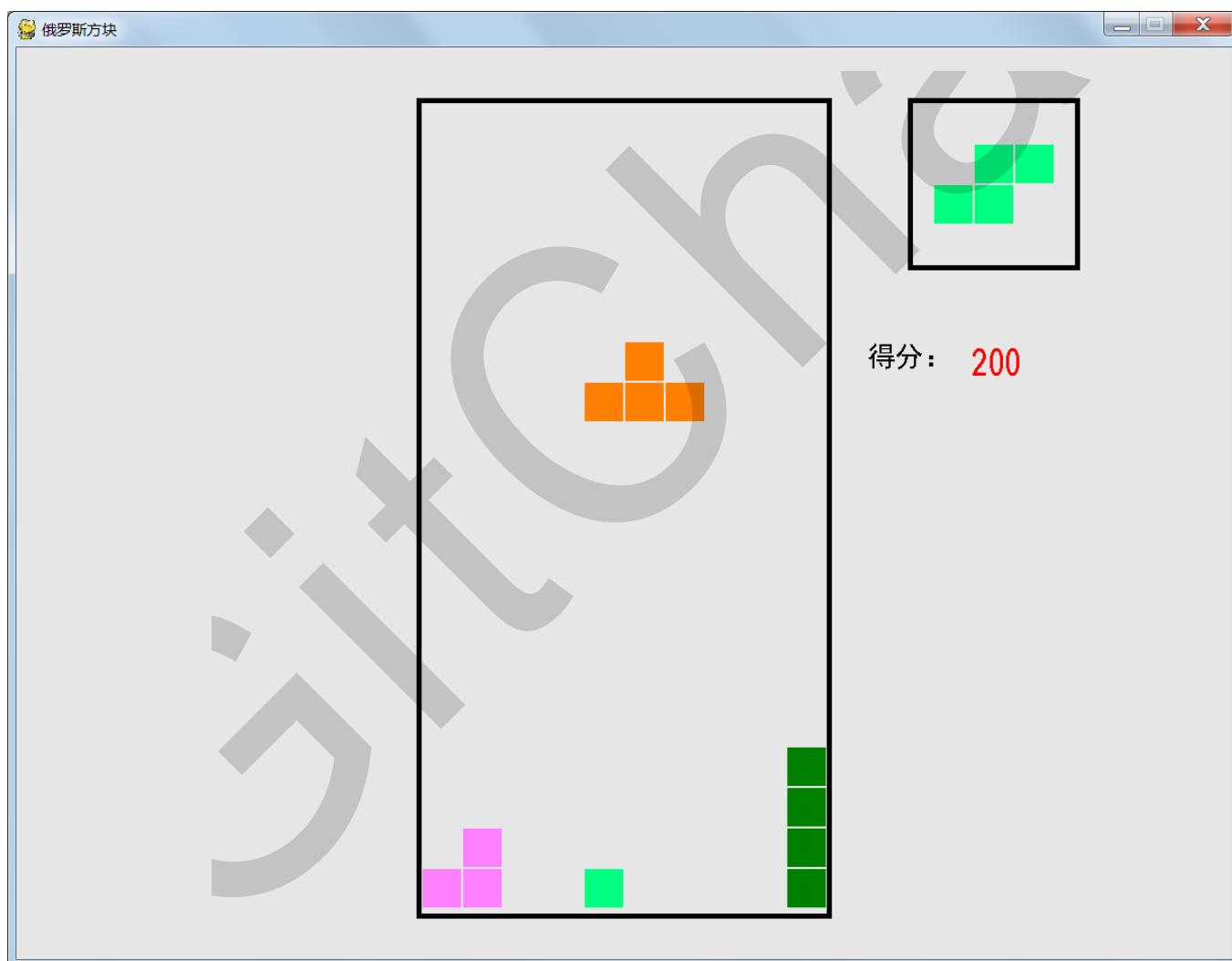


图1 去掉网格线后的界面

强制重新开始

玩家按下 r 字母键，游戏重新开始。首先，我们要在 main.py 的 `on_key_down` 函数内扩充代码来响应 r 字母键按下事件。扩充的代码如下所示。

```
1. def on_key_down(event, game_state):
2.     .....
3.     elif event.key == pygame.K_r:
4.         game_state.start_game() #按r键强制重新开始游戏
```

开始游戏函数 `start_game()` 本身也做了一点改动。这一改动是把 main.py 文件内定义的 `main` 函数中设置随机数种子的语句移到 `start_game` 方法内。

```
1. def start_game(self):
2.     self.stopped = False
3.     self.set_timer(TIMER_INTERVAL)
4.     self.timer_interval = TIMER_INTERVAL
5.     self.piece = self.new_piece() #生成第一个方块。此时
self.piece=None, self.next_piece引用方块对象。
6.     self.piece = self.new_piece() #生成第二个方块，此时self.piece引用
方块对象。
7.     self.session_count += 1
8.     self.wall.clear()
9.     self.game_score = 0
10.    self.paused = False
11.    random.seed(int(time.time())) #每次游戏，使用不同的随机数序列
```

这一改动的目的是，每轮游戏开始之际，重新产生一组随机数序列，与前一轮游戏使用的随机数序列完全不同的序列。

去掉网格线和抬高游戏区域

是时候去掉网格线了。做法是修改 `GameDisplay` 类的 `draw_game_area` 方法。如下面代码所示，注释掉的代码行是去掉的代码；下面的 `GameDisplay.draw_border` 方法是新增的；其他地方没有改动。新增代码绘制了游戏区域的边界线（见图1）。`draw_border` 方法绘制了构成边界的四个矩形（每条边都是一个矩形）。上一篇中，调用该方法绘制下一方块显示区域的边界。第二个参数决定了游戏区域左边界离窗口左边界的距离，减去 `EDGE_WIDTH` 的原因是游戏区域边界线宽为 `EDGE_WIDTH`。`EDGE_WIDTH` 常量在 `settings.py` 内定义为：`EDGE_WIDTH = 5`。

```

1.     @staticmethod
2.         def draw_game_area(screen, game_state, game_resource):
3.             '''绘制游戏区域'''
4.             # for r in range(21):
5.             #     pygame.draw.line(screen, EDGE_COLOR, (GAME_AREA_LEFT, GAM
6.             #                                     (GAME_AREA_LEFT + GAME_AREA_WIDTH,
7.             #                                     GAME_AREA_TOP + r * CELL_WIDTH))
8.             # for c in range(11):
9.             #     pygame.draw.line(screen, EDGE_COLOR, (GAME_AREA_LEFT + c
10.             #                                     * CELL_WIDTH, GAME_AREA_TOP),
11.             #                                     (GAME_AREA_LEFT + c * CELL_WIDTH,
12.             #                                     GAME_AREA_TOP + GAME_AREA_HEIGHT))
13.             GameDisplay.draw_border(screen, GAME_AREA_LEFT - EDGE_WIDTH, GA
14.             ME_AREA_TOP, LINE_NUM, COLUMN_NUM)
15.
16.             GameDisplay.draw_wall(game_state.wall)
17.             GameDisplay.draw_score(screen, game_state.game_score)
18.             if game_state.stopped:
19.                 if game_state.session_count > 0:
20.                     GameDisplay.draw_game_over(screen, game_resource)
21.                     GameDisplay.draw_start_prompt(screen, game_resource)
22.             if game_state.paused:
23.                 GameDisplay.draw_pause_prompt(screen, game_resource)
24.             GameDisplay.draw_next_piece(screen, game_state.next_piece)

```

前面，游戏区域一直紧贴窗口底部。图1中，游戏区域往上抬高了一些。我们通过改动 settings.py 文件内的 `GAME_AREA_TOP` 常量的值——减去50个像素，来抬高游戏区域。

```

1.     GAME_AREA_TOP = SCREEN_HEIGHT - GAME_AREA_HEIGHT - 50           #游戏区顶部的
    空白区的宽度

```

调整 I 型方块的颜色

在试玩俄罗斯方块过程中，我总是觉得 I 型方块的黄色太亮了。于是，把 I 型方块的颜色修改成深绿色。修改的代码位于 settings.py 文件内，即下面代码中的 `'I': (0, 128, 0)`。

```

1.     PIECE_COLORS = {
2.         'S': (0, 255, 128),
3.         'Z': (255, 128, 255),

```

```
4.     'J': (128, 0, 255),  
5.     'L': (0, 0, 255),  
6.     'I': (0, 128, 0),  
7.     'O': (255, 0, 0),  
8.     'T': (255, 128, 0)  
9. }
```

小结

本步骤实现了以下功能：

1. 玩家按下 r 字母键后，游戏重新开始。
2. 去除游戏区域的网格线，而且调整了游戏区域在窗口内的位置。

你可从下面的链接浏览或下载代码：

- [Github](#)

随着代表不同功能的字母按键种类越来越多，玩家不容易记住它们。下一篇将提供使用指南，还考虑增加难度级别的功能。