第15课：消行计分

步骤目标

本文要实现的目标是：当方块触底时，如果游戏区域的某一行或某几行满行了（即该行各列单元格全部被方块填充），那么消掉这些行，并计分。这组动作简称为“消行计分”。图1中，底部倒数第2行将被消行，并得到100分。

第二个目标是在游戏窗口的右上角显示得分。一开始，得分为0。

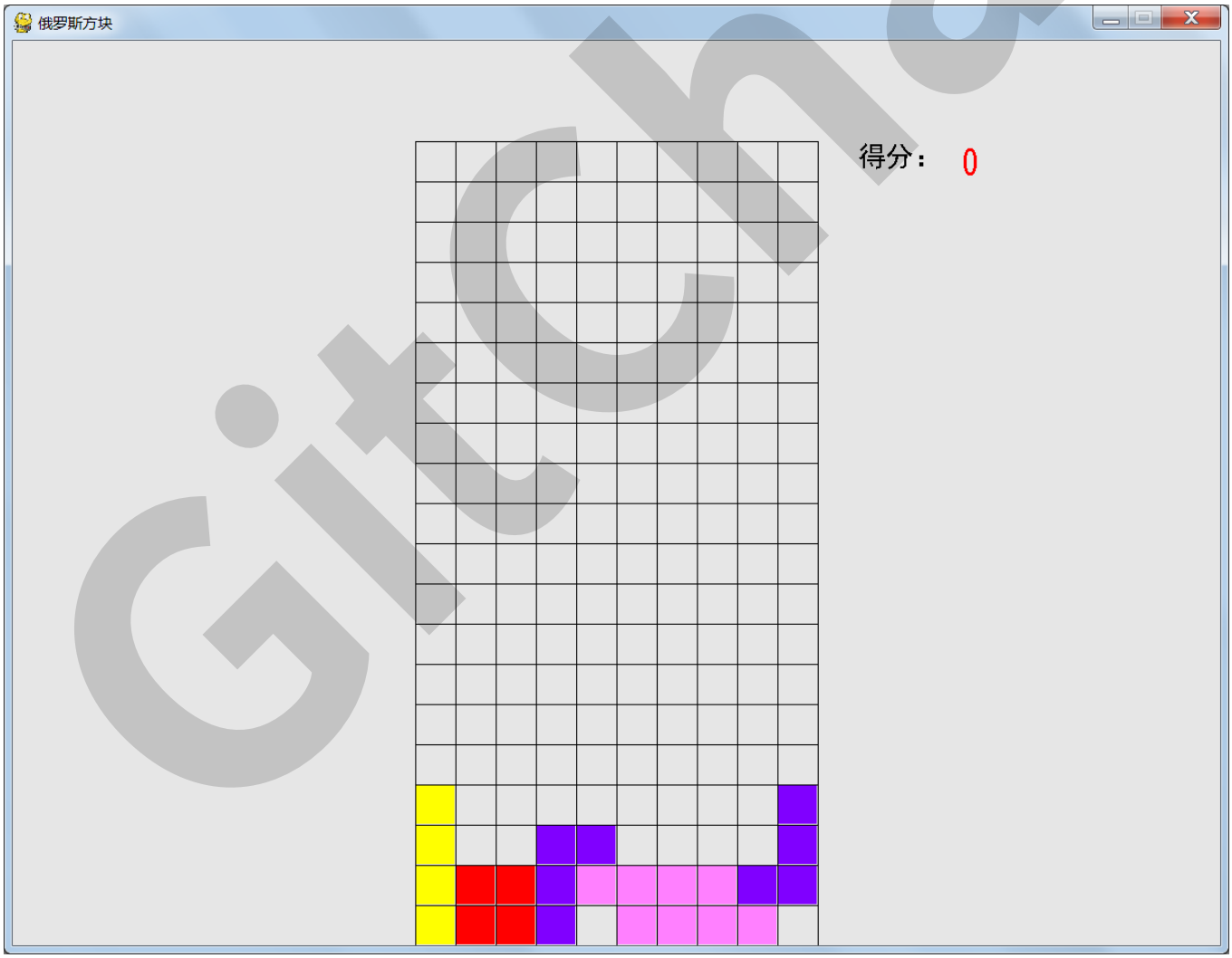

图1 消行计分

Github

本步骤的任务是：

1. 每一次方块触底时，发起消行计分动作。
2. 消行并得出本次消行的得分。
3. 累计游戏进行过程中的总得分。
4. 显示总得分。

发起消行计分动作

当前方块触底时，发起消行计分动作。消行计分的代码如代码1中第33行所示。这里，修改的代码文件是 main.py。第32行是把当前方块砌入墙体，接下来就是发起消行计分了。

```
1.  # TetrisGame/main.py
2.  15  def main():
3.  …       ……  #与前一版本相同，故省略。
4.  27      game_state = GameState(screen)
5.  28      #游戏主循环
6.  29      while True:
7.  30          #方块触底的话
8.  31          if game_state.piece.is_on_bottom:
9.  32              game_state.wall.add_to_wall(game_state.piece)
10. 33              game_state.add_score(game_state.wall.eliminate_lines())
11. 34              # print(game_state.game_score)
12. 35              game_state.piece = Piece(random.choice(PIECE_TYPES), sc
    reen, game_state.wall)
13. 36          ……  #与前一版本相同，故省略代码。
```

代码1 发起消行计分的代码

消行计分

消行计分功能由 GameWall 类来实现，因为墙体对象的墙体矩阵记录了哪些单元格有砖块。

消行计分的做法是：

1. 扫描墙体矩阵，得出要被消掉的一行或多行（最多4行）。
2. 对上一小步找出来的每一行，分别进行消行。细节在下面讲解。
3. 根据消掉的行数计分。

GitChat

GameWall 类的 `eliminate_lines` 方法完成以上功能，如代码2所示。这一方法位于 gamewall.py 文件的第41~72行。

```
# TetrisGame/gamewall.py
41  def eliminate_lines(self):
42      '''消行。如果一行没有空白单元格，就消掉该行。返回得分。'''
43      '''
44      计分规则：
45      消掉0行：0分
46      消掉1行：100分
47      消掉2行：200分
48      消掉3行：400分
49      消掉4行：800分
50      '''
51      #需要消哪几行
52      lines_eliminated = [ ]
53      for r in range(LINE_NUM):
54          if self.is_full(r):
55              lines_eliminated.append(r)
56
57      #消行，更新墙体矩阵
58      for r in lines_eliminated:
59          self.copy_down(r)   #消掉行r，上面的各行依次下沉一行。
60          for c in range(COLUMN_NUM):
61              self.area[0][c] = WALL_BLANK_LABEL
62
63      #根据消掉的行数，计算得分
64      eliminated_num = len(lines_eliminated)
65      assert(eliminated_num <= 4 and eliminated_num >= 0)
66      if eliminated_num < 3:
67          score = eliminated_num * 100
68      elif eliminated_num == 3:
69          score = 400
70      else:
71          score = 800
72      return score
```

代码2 消行计分的代码

对于以上代码，简要说明如下：

1. 第52~55行得出要消掉的行，把行号存入列表 `lines_eliminated`。此处调用的

GitChat

`is_full(r)` 是 GameWall 类的方法，作用是看下标为 r 的一行是否满行。如果满了，`is_full` 方法返回 True，否则返回 False。代码如下：

```
def is_full(self, row):
    '''下标为row的一行满了吗'''
    for c in range(COLUMN_NUM):
        if self.area[row][c] == WALL_BLANK_LABEL:
            return False

    return True
```

2. 代码2中第58~61行完成消行功能。怎么消掉某一行，比如第 r 行呢？做法是，把第 r 行上面的各行，从下到上依次下沉一行。也就是说，把第 r-1 行下沉到第 r 行，第 r-2 行下层到第 r-1 行，依次类推。最后，在第1行（行下标为0），补上一空白行。GameWall 类的 `copy_down()` 方法实现消除一行功能，代码见下。

```
def copy_down(self, row):
   '''把row号行上面各行依次下沉一行。'''
   for r in range(row, 0, -1):
       for c in range(COLUMN_NUM):
          self.area[r][c] = self.area[r - 1][c]
```

3. 第64~71行完成计分功能。计分规则如注释所列。我们可以把计分功能从 `eliminate_lines` 方法内剥离出去，独立作为一个 GameWall 类的方法。

4. 第72行，返回本次消行的得分。

统计总得分

游戏总得分属于游戏状态的组成部分，因此在 GameState 类内安排一个属性 `game_score` 记录总得分，安排一个方法 `add_score` 来累计总得分，如代码3所示。代码1中第33行调用了 `add_score` 方法，以统计总得分。调用 `add_score` 使用的参数是 `game_state.wall.eliminate_lines()`，后者将执行消行计分处理，返回本次消行的得分。

```
# TetrisGame/gamestate.py
10  class GameState():
11      def __init__(self, screen):
```

GitChat

```
12          self.screen = screen
13          self.wall = GameWall(screen)
14          self.piece = Piece(random.choice(PIECE_TYPES), screen, self.wall)
15          self.timer_interval = TIMER_INTERVAL   #1000ms
16          self.set_timer(self.timer_interval)
17          self.game_score = 0
18
19      def set_timer(self, timer_interval):
20          self.game_timer = pygame.time.set_timer(pygame.USEREVENT, timer_interval)
21
22      def add_score(self, score):
23          self.game_score += score
```

代码3 统计总得分

显示得分

技术角度讲，显示得分的要点是在游戏窗口上显示文字。Pygame 提供一组显示文字的函数。我们在 GameDisplay 类内定义 `draw_score` 方法来完成显示得分的功能，如代码4所示。GameDisplay 类的 `draw_game_area` 方法尾部，调用 `draw_score` 方法。`draw_game_area` 方法则被程序主循环调用，因此每循环一次就会刷新窗口，得分也会被刷新。

```
# TetrisGame/gamedisplay.py
40  @staticmethod
41  def draw_score(screen, score):
42      '''绘制游戏得分'''
43      score_label_font = pygame.font.SysFont('simhei', 28)    #换成'arial', 无法显示中文。
44
45      score_label_surface = score_label_font.render(u'得分：', False, SCORE_LABEL_COLOR)
46      score_label_position = (GAME_AREA_LEFT + COLUMN_NUM * CELL_WIDTH + 40, GAME_AREA_TOP)
47      screen.blit(score_label_surface, score_label_position)
48
49      score_font = pygame.font.SysFont('arial', 36)
50      score_surface = score_font.render(str(score), False, SCORE_COLOR)
51      score_label_width = score_label_surface.get_width()
```

```
14.  52      score_position = (score_label_position[0] +score_label_width +
     20, score_label_position[1])
15.  53      screen.blit(score_surface, score_position)
```

代码4 显示得分的 `draw_score` 方法（左侧数字是文件内代码行号）

对于上述代码，简要说明如下：

1. 第43行代码是调用 SysFont 函数生成一个字体对象 `score_label_font`。第一个参数指定字体，这里是 simhei（即黑体）。第二个参数指定字体大小，这里是28。
2. 第45行代码是调用字体对象的 render 方法生成图元对象 `score_label_surface`。render 方法的第一个参数指定要绘制的文字，单引号前面的 u 字母表示这是 Unicode 型字符串。字符串内有中文的话，建议你用 u 字母标示出来。第二个参数是指是否启用反显效果。False 是指不启用。第三个参数是指文字的颜色。这里，`SCORE_LABEL_COLOR` 常量定义为 `(0, 0, 0)`，即黑色。
3. 第46行设定了文字图元显示的位置。
4. 第47行的作用是在游戏窗口内显示文字图元。
5. 到第47行代码，只是显示了“得分：”这个标签，参见图1。接下来要显示分数。做法上，把上述第1~4点重来一遍，不再赘述。
6. GameDisplay 类的 `draw_game_area` 方法尾部，调用 `draw_score` 方法。

> 注意：在 Windows 系统中，代码4能够正常运行。如果你是在 Linux 系统中编写程序，那么有可能因为系统没有安装 simhei（即黑体）这种字体而导致程序运行错误。错误发生在代码4中的第43行。纠正该错误需要安装字体文件。做法参阅以下文档，你也可以自行搜索“如何在 Linux 中安装字体”的答案。在 Mac 系统中遇到字体问题，请自行搜索答案。
>
> 1. 《Linux CentOS 7 安装字体库 & 中文字体》：该文适用于 Redhat 或 CentOS 系统。文内详细讲解了：（1）安装字体库的方法；（2）在 Windows 系统内找到字体文件和复制到 Linux 系统的字体文件目录的做法；（3）配置字体的做法。
> 2. 《Linux 命令之 FC，手动安装字体》：该文适用于 Redhat 或 CentOS 系统，比上一文档更简练，信息更少，不过步骤是完整的。可与上一文档配合使用。
> 3. 《Linux 中安装字体》：该文适用于 Ubuntu 系统。它扼要地讲解了操作步骤。第4步加载字体中的第三条命令 `fc -cache -fv`，应改为 `fc-cache -fv`，`fc-cache` 是命令。建议与第一份文档配合使用。

GitChat

小结

本文实现了消行计分功能。做法是：

1. 在程序主循环内，一检测到方块触底，就调用墙体对象的 `eliminate_lines` 方法，然后把该方法返回的本次消行的得分累加到游戏状态对象的 `game_score` 属性上。如果没有消掉任何行，那么也会累加分数，只不过是加上0分。
2. 墙体对象的 `eliminate_lines` 方法内部，首先扫描墙体矩阵，得出要消掉哪些行；然后采用逐行下层的方式消行；最后得出本次消行的得分，并返回得分。
3. 游戏状态对象的 `game_score` 属性记录本轮游戏的总得分。
4. 显示得分的执行流程是：程序主循环调用 GameDisplay 类的 `draw_game_area` 方法，后者调用 `draw_score` 方法。每执行一次主循环，将刷新整个窗口，游戏得分也会被刷新。

可到以下链接浏览或下载完成本实验步骤全部功能的代码，仅供参考：

- Github

到此，我们的游戏值得玩一玩了。玩几把看看，你能连续得多少分？重点是，你是这款游戏的创作者。

下一篇中，我们将实现“按 s 键开始游戏”功能。目前，程序一启动，游戏就开始了。我们应该让玩家自己来开始游戏。

GitChat