

第19课：防止穿墙

步骤目标

本文的目标是防止穿墙。

到上一篇为止，我们实现的代码版本会发生穿墙的问题。这一问题分以下情形：

1. 方块向右移动，会穿墙。如图1所示，Z 型方块向右移动穿过墙体，即 J 型方块垒砌的部分。
2. 方块向左移动，会穿墙。
3. 方块翻转，会穿墙。
4. 新方块刚出现在游戏区域顶部，会穿墙。对于这一情形，我们不阻止穿墙行为，而是报告游戏结束。

你自己可以运行程序试一下，是否会发生穿墙现象。值得指出的是，向下移动不会穿墙。我们从中可以借鉴一点做法。

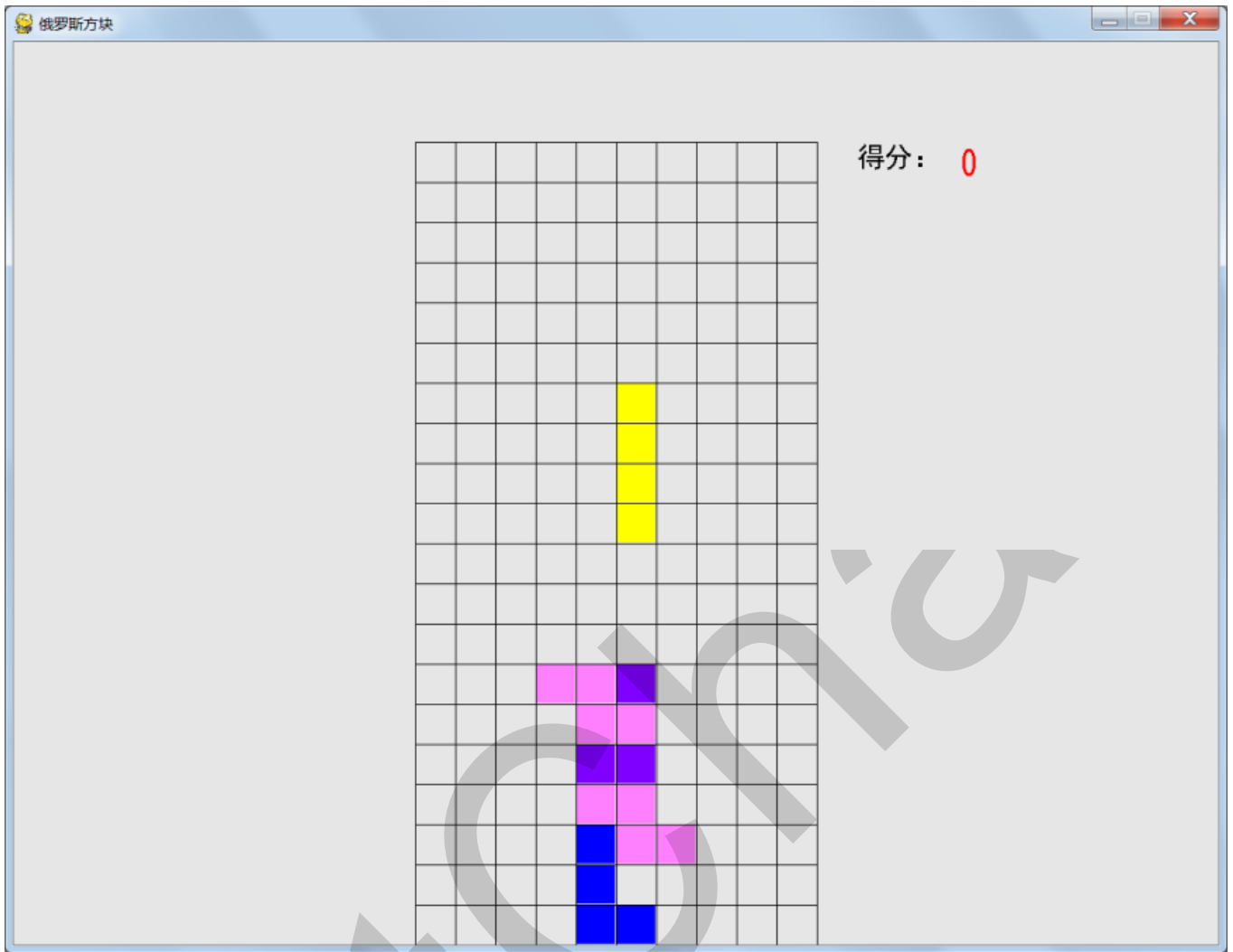


图1 方块向右移动会穿墙

防止穿墙

为什么方块向下移动不会穿墙呢？这是因为判断方块能否向下移动的方法 `can_move_down()` 做了点特别的事情，就是检测紧挨着方块的下方单元格有没有被墙体占据。看以下代码，`or self.game_wall.is_wall(self.y + r + 1, self.x + c)` 这句代码用于防止向下移动时穿墙。

```
1. def can_move_down(self):
2.     shape_mtx = PIECES[self.shape][self.turn_times] #姿态矩阵
3.     # print(shape_mtx)
4.     for r in range(len(shape_mtx)):
5.         for c in range(len(shape_mtx[0])):
6.             if shape_mtx[r][c] == 'O':
```

```

7.         if self.y + r >= LINE_NUM - 1 or self.game_wall.is_wall
           (self.y + r + 1, self.x + c):
8.             return False
9.     return True

```

判别方块能否向右移动的 `can_move_right` 也该这么做。下面代码是 `can_move_right` 方法的定义。`or self.game_wall.is_wall(self.y + r, self.x + c + 1)` 是新增的，作用是检测紧挨着方块的右侧单元格有没有墙体。

```

1.     def can_move_right(self):
2.         shape_mtx = PIECES[self.shape][self.turn_times] #姿态矩阵
3.         # print(shape_mtx)
4.         for r in range(len(shape_mtx)):
5.             for c in range(len(shape_mtx[0])):
6.                 if shape_mtx[r][c] == 'O':
7.                     if self.x + c >= COLUMN_NUM - 1 or
self.game_wall.is_wall(self.y + r, self.x + c + 1):
8.                         return False
9.     return True

```

判别方块能否向左移动的 `can_move_left` 方法

内，`or self.game_wall.is_wall(self.y + r, self.x + c - 1)` 为新增代码片段。新增代码的作用是检测紧挨着方块的左侧有没有墙体。

```

1.     def can_move_left(self):
2.         shape_mtx = PIECES[self.shape][self.turn_times] #姿态矩阵
3.         # print(shape_mtx)
4.         for r in range(len(shape_mtx)):
5.             for c in range(len(shape_mtx[0])):
6.                 if shape_mtx[r][c] == 'O':
7.                     if self.x + c <= 0 or self.game_wall.is_wall(self.y
+ r, self.x + c - 1):
8.                         return False
9.     return True

```

判断方块能否翻转的 `can_turn` 方法改成下面的写

法。`or self.game_wall.is_wall(self.y + r, self.x + c)` 是修改的内容。作用是检测翻转到的单元格有没有墙体。

```

1.     def can_turn(self):
2.         shape_list_len = len(PIECES[self.shape])
3.         turn_times = (self.turn_times + 1) % shape_list_len
4.         shape_mtx = PIECES[self.shape][turn_times]
5.         for r in range(len(shape_mtx)):
6.             for c in range(len(shape_mtx[0])):
7.                 if shape_mtx[r][c] == 'O':
8.                     if (self.x + c < 0 or self.x + c >= COLUMN_NUM) or (
9. self.y + r < 0 or self.y + r >= LINE_NUM) \
10.                                     or self.game_wall.is_wall(self.y + r, self.x
11. + c):
12.                                     return False
13.         return True

```

由于发生修改的代码行太长，该代码行分成了两行。这两行中的第一行末尾用了反斜杠。这里，反斜杠是跨行连接符，作用是在语法上把两行连接成一行。效果上，这等同于第二行接在第一行末尾。

以上修改防止穿墙行为的发生。你可以再次运行程序测试看看。

新方块刚出现时撞墙

当游戏程序生成新方块，该方块刚出现在游戏区域顶部时撞墙，我们应当视为游戏结束。我们要修改 `GameState` 类的 `touch_bottom` 方法来做到这一点。修改后的 `touch_bottom` 方法如下所示，代码最后四行是修改过的部分。

```

1.     def touch_bottom(self):
2.         self.wall.add_to_wall(self.piece)
3.         self.add_score(self.wall.eliminate_lines())
4.         for c in range(COLUMN_NUM):
5.             if self.wall.is_wall(0, c):
6.                 self.stopped = True
7.                 break
8.         if not self.stopped:
9.             self.piece = Piece(random.choice(PIECE_TYPES), self.screen, self.wall)
10.             if self.piece.hit_wall():
11.                 self.stopped = True
12.         if self.stopped:
13.             self.stop_timer()

```

上面所调用的 `hit_wall` 方法在 `Piece` 类中定义，代码如下。

```
1. def hit_wall(self):
2.     shape_mtx = PIECES[self.shape][self.turn_times]
3.     for r in range(len(shape_mtx)):
4.         for c in range(len(shape_mtx[0])):
5.             if shape_mtx[r][c] == 'O':
6.                 if self.game_wall.is_wall(self.y + r, self.x + c):
7.                     return True
8.     return False
```

小结

本文解决了以下穿墙问题：

1. 向右移动时，方块能穿墙。对策是在移动前检测紧挨着方块的右侧单元格有没有墙体。
2. 向左移动时，方块能穿墙。对策是在移动前检测紧挨着方块的左侧单元格有没有墙体。
3. 翻转时，方块能穿墙。对策是翻转前检测翻转到的单元格有没有墙体。
4. 生成新方块的时候，新方块出现在游戏区域顶部中央。此时，方块可能会穿墙。如果发生穿墙的情形，意味着游戏结束。

你可以从以下链接浏览或下载完成本步骤的全部功能的代码。

- [Github](#)

下一篇要实现的功能是“提示下一方块”，即在游戏窗口右上角显示下一方块是什么。借助这一提示，玩家能规划消行的策略，更快地得到高分。