

第17课：玩家暂停游戏

步骤目标

本文的目标是实现暂停游戏功能。玩家在游戏进行中，按下 p 字母键，游戏暂停，如图1所示。玩家再次按下 p 字母键，游戏继续。

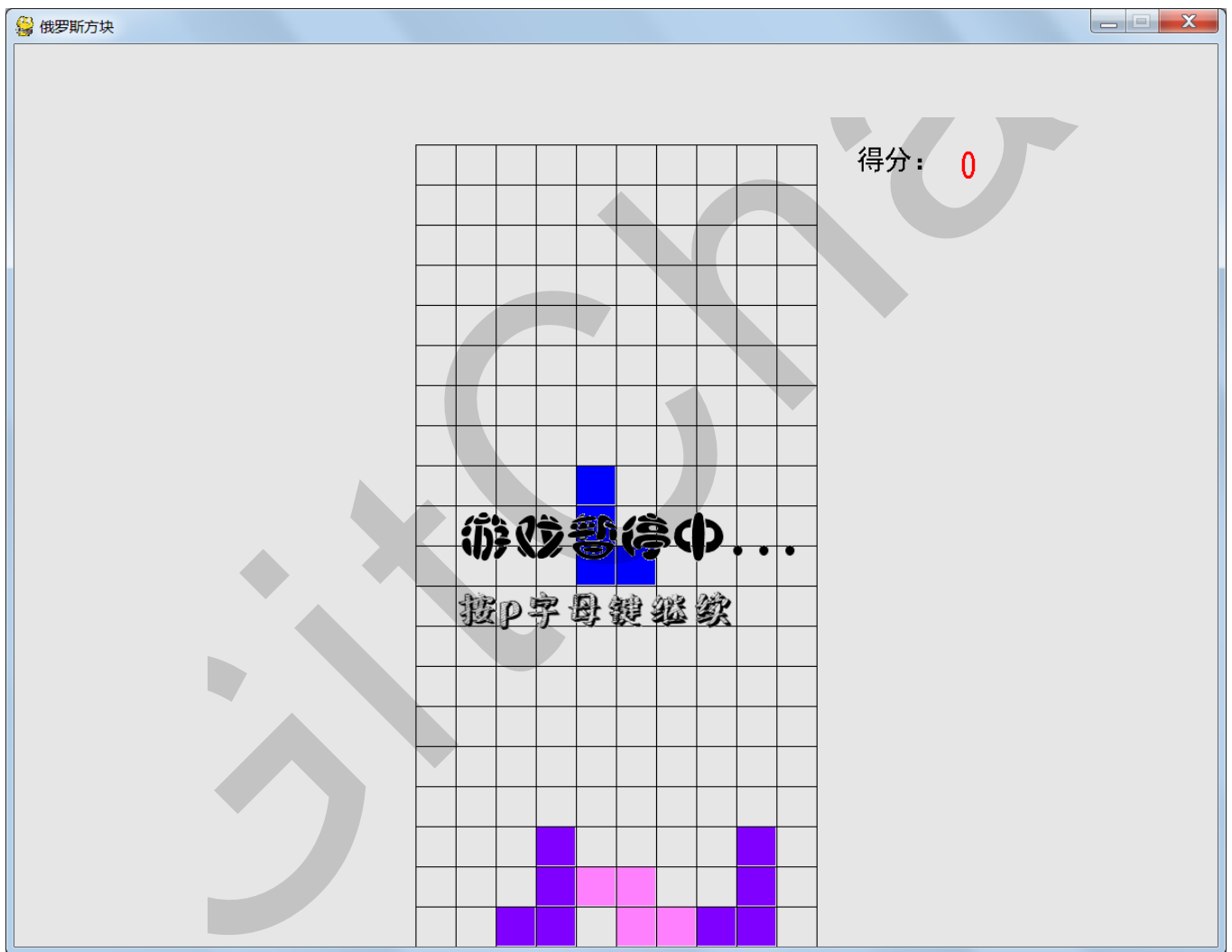


图1 游戏暂停中

本步骤的任务包括：

1. 响应 p 字母键按下事件，暂停游戏。暂停游戏的动作是取消定时器。

2. 显示“游戏暂停中...”和“按 p 字母键继续”的提示。
3. 玩家再次按 p 字母键的话，继续游戏。继续游戏的动作是启动定时器，并清除上述提示。

暂停游戏和继续游戏

玩家按下 p 字母键，游戏暂停。这意味着要响应处理键盘按键事件。在 main.py 文件内，`on_key_down()` 函数的末尾加几行代码即可响应 p 字母键按下事件，见代码1第86~90行代码。

```
1. # TetrisGame/main.py
2. 64 def on_key_down(event, game_state):
3.     ..... #与前一版本相同，故省略。
4. 86 elif event.key == pygame.K_p and not game_state.stopped:
5. 87     if game_state.paused:
6. 88         game_state.resume_game()
7. 89     else:
8. 90         game_state.pause_game()
```

代码1 响应 p 字母键按下事件的代码

对代码1中的代码，简要说明如下：

1. 第86行中“not `game_state.stopped`”这一条件不能丢。这样，在游戏停止时，不会去理睬 p 字母键按下事件。
2. 收到 p 字母键按下事件后，如果游戏处于暂停状态，则表明要继续游戏；如果游戏处于进行中状态，则表明要暂停游戏。继续游戏调用游戏状态对象 `game_state` 的 `resume_game()` 方法。暂停游戏调用 `pause_game()` 方法。
3. `resume_game()` 和 `pause_game()` 都在 `GameState` 类内定义。此外，`GameState` 类还要增加 `paused` 属性，当属性值为 `True` 时，表明游戏处于暂停状态，否则游戏处于进行中状态。这些定义如代码2所示。

```
1. # TetrisGame/gamestate.py
2. 10 class GameState():
3. 11     def __init__(self, screen):
4. 12         self.screen = screen
5. 13         self.wall = GameWall(screen)
6. 14         self.piece = None
7. 15         self.timer_interval = TIMER_INTERVAL #1000ms
```

```

8.   16         self.game_score = 0
9.   17         self.stopped = True
10.  18         self.paused = False
11.  ...         ..... #其他方法的定义
12.  32     def pause_game(self):
13.  33         pygame.time.set_timer(pygame.USEREVENT, 0) #传入0表示清除定
           时器
14.  34         self.paused = True
15.  35
16.  36     def resume_game(self):
17.  37         self.set_timer(self.timer_interval)
18.  38         self.paused = False

```

代码2 暂停游戏的相关代码

从代码2可见，暂停游戏方法 `pause_game` 的动作有两个：

1. 取消定时器。
2. 把登记暂定状态的属性 `paused` 设为 `True`。

`resume_game` 方法的动作恰好相反。

暂停的提示

我们从图1可以看到，游戏暂停时，游戏区域中央位置显示“游戏暂停中...”和“按 p 字母键继续”的提示。下面讲如何做到这件事。这与上一文所实现的显示“按 s 字母键开始”的做法是类似的。

首先，你要把“游戏暂停中...”图片文件和“按 p 字母键继续”图片文件制作好。你也可以直接从以下链接下载它们。我是利用[字体转换器网站](#)来制作图片的。

- 图片文件下载：[Github](#)

图片文件放到项目文件夹的 `images` 子文件内，如图2所示。其中，`game_pausing.png` 是“游戏暂停中...”图片文件，而 `press-p-continue.png` 是“按 p 字母键继续”图片。

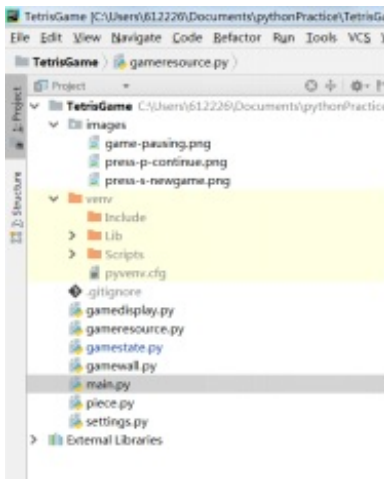


图2 暂停功能使用的图片文件

接着，在 GameResource 类中扩充加载图片的代码，如代码3所示。

```
1.  # TetrisGame/gameresource.py
2.  6   class GameResource():
3.  7       def __init__(self):
4.  8           self.img_path = 'images/'
5.  9           self.newgame_img = None
6.  10          self.pausing_img = None
7.  11          self.continue_img = None
8.  12
9.  13      def load_newgame_img(self):
10. 14          if not self.newgame_img:
11. 15              self.newgame_img = pygame.image.load(self.img_path + "p
ress-s-newgame.png").convert_alpha()
12. 16              return self.newgame_img
13. 17
14. 18      def load_pausing_img(self):
15. 19          if not self.pausing_img:
16. 20              self.pausing_img = pygame.image.load(self.img_path + "g
ame-pausing.png").convert_alpha()
17. 21              return self.pausing_img
18. 22
19. 23      def load_continue_img(self):
20. 24          if not self.continue_img:
21. 25              self.continue_img = pygame.image.load(self.img_path + "
press-p-continue.png").convert_alpha()
22. 26              return self.continue_img
```

代码3 加载图片的代码

简要说明代码3中的代码:

1. 第10行, `pausing_img` 属性用于引用“游戏暂停中...”图片对象。图片文件加载到内存后, 成为图片对象。
2. 第11行, `continue_img` 属性用于引用“按 p 字母键继续”图片对象。
3. 第18~21行, 加载“游戏暂停中...”图片。如果此前已经加载过, 则不重复加载。load 函数的参数是图片文件的路径。 `convert_alpha` 函数的作用是生成具有 Alpha 通道, 即透明效果的图片对象。
4. 第23~26行, 与第3点一样。

最后, 在 `gamedisplay.py` 文件内扩充代码实现游戏暂停期间调用加载函数绘制图片, 如代码4所示。代码4中, 第33-34行、第66-72行是新增的代码。第一项扩充是 `GameDisplay` 类的 `draw_game_area` 方法尾部, 新增了两行代码, 即第33、34行代码。这两行代码的作用是, 如果游戏处于暂停状态, 就调用 `draw_pause_prompt` 方法显示暂停提示信息, 效果如图1所示。

第66~72行定义了 `draw_pause_prompt` 方法。方法体内, 第一次调用 `screen.blit` 用于显示“游戏暂停中...”图片; 第二次调用 `screen.blit` 用于显示“按 p 字母键继续”图片。你可以在搜索引擎中输入“pygame blit”来查询 blit 函数的作用。

```
1.     # TetrisGame/gamedisplay.py
2.     19     @staticmethod
3.     20     def draw_game_area(screen, game_state, game_resource):
4.     ...         ..... #与前一版本相同, 故省略
5.     33         if game_state.paused:
6.     34             GameDisplay.draw_pause_prompt(screen, game_resource)
7.
8.     ...
9.     ..... #其他方法的定义
10.    65     @staticmethod
11.    66     def draw_pause_prompt(screen, game_resource):
12.    67         '''显示游戏暂停'''
13.    68         pause_position = (GAME_AREA_LEFT + 1 * CELL_WIDTH,
14.    GAME_AREA_TOP + 9 * CELL_WIDTH)
15.    69         screen.blit(game_resource.load_pausing_img(), pause_position)
16.    70
17.    71         resume_tip_position = (GAME_AREA_LEFT + 1 * CELL_WIDTH, GAME_AR
18.    EA_TOP + 11 * CELL_WIDTH)
19.    72         screen.blit(game_resource.load_continue_img(),
```

```
resume_tip_position)
```

代码4 显示暂停提示信息

暂停期间不能响应方向键

游戏暂停了，玩家按上下左右方向键的话，程序应不予理睬。代码5和代码6所列出的，在 main.py 文件的 `on_key_down` 函数内的改动做到了这一点。代码5是改动前的版本，代码6是改动后的版本。代码6中的改动之处请看代码中的注释说明。

```
1. def on_key_down(event, game_state):
2.     if event.key == pygame.K_DOWN:
3.         # print("向下方向键被按下")
4.         if game_state.piece:
5.             game_state.piece.move_down()
6.     elif event.key == pygame.K_UP:
7.         # print("向上方向键被按下")
8.         if game_state.piece:
9.             game_state.piece.turn()
10.    elif event.key == pygame.K_RIGHT:
11.        # print("向右方向键被按下")
12.        if game_state.piece:
13.            game_state.piece.move_right()
14.    elif event.key == pygame.K_LEFT:
15.        # print("向左方向键被按下")
16.        if game_state.piece:
17.            game_state.piece.move_left()
18.    elif event.key == pygame.K_f:
19.        if game_state.piece:
20.            game_state.piece.fall_down()
21.    elif event.key == pygame.K_s and game_state.stopped:
22.        game_state.start_game()
```

代码5 游戏暂停期间不响应方向键（修改前）

```
1. def on_key_down(event, game_state):
2.     if not game_state.paused and event.key == pygame.K_DOWN: # 此处有修
    改
3.         # print("向下方向键被按下")
4.         if game_state.piece:
5.             game_state.piece.move_down()
```

```

6.         elif not game_state.paused and event.key == pygame.K_UP: # 此处有修
           改
7.             # print("向上方向键被按下")
8.             if game_state.piece:
9.                 game_state.piece.turn()
10.        elif not game_state.paused and event.key == pygame.K_RIGHT: # 此处
           有修改
11.            # print("向右方向键被按下")
12.            if game_state.piece:
13.                game_state.piece.move_right()
14.        elif not game_state.paused and event.key == pygame.K_LEFT: # 此处有
           修改
15.            # print("向左方向键被按下")
16.            if game_state.piece:
17.                game_state.piece.move_left()
18.        elif not game_state.paused and event.key == pygame.K_f: # 此处有修改
19.            if game_state.piece:
20.                game_state.piece.fall_down()
21.        elif event.key == pygame.K_s and game_state.stopped:
22.            game_state.start_game()
23.        elif event.key == pygame.K_p and not game_state.stopped:
24.            if game_state.paused:
25.                game_state.resume_game()
26.            else:
27.                game_state.pause_game()

```

代码6 游戏暂停期间不响应方向键（修改后）

小结

本文步骤实现玩家按下 p 字母键暂停或继续游戏的功能。做法是：

1. 在 `on_key_down` 函数内添加响应 p 字母键按下事件。要注意一个细节，当游戏处于停止状态的情况下，不能响应 p 字母键按下事件。
2. 在 `GameState` 类内添加 `paused` 属性，值为 `True` 是表明处于暂停状态，为 `False` 则意味着不处于暂停状态。又添加 `pause_game` 方法和 `resume_game` 方法。前者暂停游戏，后者继续游戏。响应 p 字母键按下事件时，调用两者之一。
3. 在 `GameDisplay` 类内添加静态方法 `draw_pause_prompt`。该方法的作用是在游戏暂停的时候，显示“游戏暂停”和“按 p 字母键继续”。程序主循环调用 `draw_game_area` 函数。后者会检测游戏的运行状态，若是暂停状态，则调用 `draw_pause_prompt`。

4. 清除副作用。游戏暂停之际，玩家按下任何方向键，都要忽略该事件。

完成本步骤全部功能的代码可从以下链接浏览或下载。

- [Github](#)

下一步骤，我们实现结束游戏的功能。以后还会增加音效和背景图片。我们的游戏将越来越高大了。