

第06课：捕捉键盘按键事件

步骤目标

本文将实现的目标是：

1. 程序运行期间，捕捉到键盘上下方向键、左右方向键被按下的事件，并报告是哪个键被按下。
2. 定义 `check_events` 函数来处理键盘按键事件。

本步骤完成后，达成的效果如图1所示。下一篇课程中，我们将通过按键来移动方块。

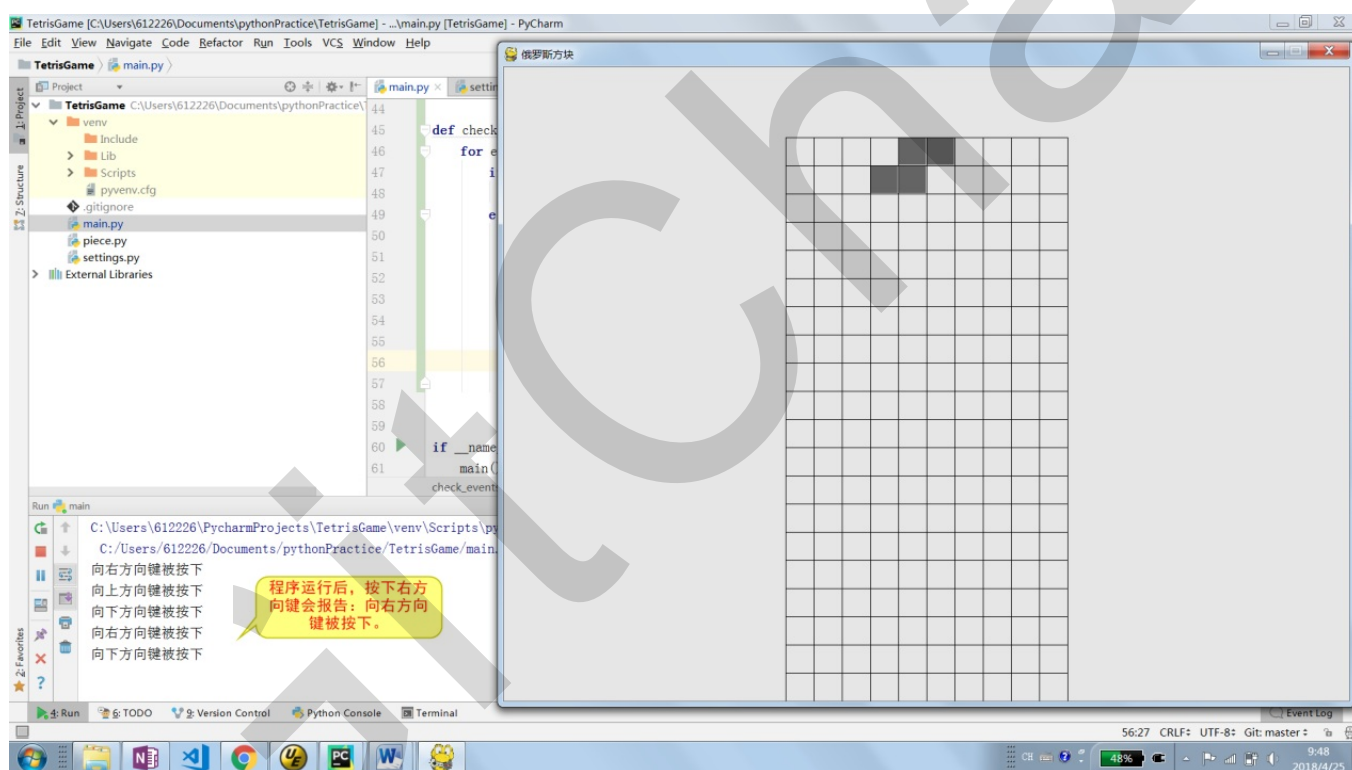


图1 处理键盘按键事件

什么时候捕捉键盘按键

程序运行期间，玩家按下键盘上的某个键，将产生键盘按键事件（Event）。我们的游戏程序利用 `pygame.event` 对象来捕捉键盘按键事件。

我们要在程序的主循环中捕捉键盘按键事件，如代码1所示。具体而言，就是在 `main.py` 文件

的 main 函数内，进一步地，是在 while 循环内。

```
1.  # TetrisGame/main.py
2.  10 def main():
3.  11     #初始化pygame。
4.  12     pygame.init()
5.  ...     ..... #省略部分代码
6.  21
7.  22     #游戏主循环
8.  23     while True:
9.  24         #监视键盘和鼠标事件
10.  25         for event in pygame.event.get():
11.  26     if event.type == pygame.QUIT:
12.  27         sys.exit()
13.  28         #此处捕捉键盘按键事件
14.  ...         ..... #省略绘制屏幕的代码
```

代码1 主循环中捕捉键盘按键事件

代码1中，第25~27行代码的作用是响应关闭窗口事件而后关闭程序。这在以前的开发步骤中就使用了。这里对这三行代码稍作解释。

pygame.event.get() 的作用是从事件队列内取出全部事件对象，得到待处理事件列表。玩家按下键盘或者点击关闭窗口口的按钮，都会产生事件。新发生一个事件，就会生成 pygame.event 事件对象，并把该事件对象加入到事件队列尾部。调用一次 pygame.event.get()，效果上就等同于从事件队列中取出全部事件对象，得到待处理事件列表。这样，下面的代码：

```
1.  for event in pygame.event.get()
```

每循环一次，event 变量被赋值为待处理事件列表中的下一个事件对象。

事件对象 event 有一个属性叫做 type，它用于判别事件的类型。事件类型有很多种，这里我们只简要介绍三种事件类型。

1. **关闭窗口事件类型。** 用常量 pygame.QUIT 来标识这种类型。如果

`event.type == pygame.QUIT` 为真的话，意味着程序收到了关闭窗口事件。代码1的第27行代码的作用是退出程序——这正是关闭窗口事件的响应处理。

2. **键盘按下事件类型。** 用常量 `pygame.KEYDOWN` 来标识这种类型。如果

`event.type == pygame.KEYDOWN` 为真的话，意味着程序收到了键盘上任意一个按键按下
的事件。

3. **键盘松开事件类型。** 用常量 `pygame.KEYUP` 来标识这种类型。如果

`event.type == pygame.KEYUP` 为真的话，意味着程序收到了先前按下的键松开的事件。

有人会问，为啥要区分按下和松开两种情形呢？这是为了达成精细的控制。我们的俄罗斯
方块程序只需要响应键盘按下事件，但有的程序还需要响应按键松开事件。

要捕捉键盘按键事件，我们需要在以下 `for` 循环内插入代码：

```
1. for event in pygame.event.get():
2.     ..... #捕捉键盘按键事件
```

如代码1第28行所示。

事件响应函数 `check_events`

为避免主循环代码越来越长，我们用一个函数来封装事件的响应处理。这个函数命名为
`check_events`，定义如代码2所示。这个函数在 `main.py` 文件内定义，开头是第45行。

```
1. # TetrisGame/main.py
2. 45 def check_events():
3. 46     '''捕捉和处理键盘按键事件'''
4. 47     for event in pygame.event.get():
5. 48         if event.type == pygame.QUIT:
6. 49             sys.exit()
7. 50         elif event.type == pygame.KEYDOWN:
8. 51             if event.key == pygame.K_DOWN:
9. 52                 print("向下方向键被按下")
10. 53             elif event.key == pygame.K_UP:
11. 54                 print("向上方向键被按下")
12. 55             elif event.key == pygame.K_RIGHT:
13. 56                 print("向右方向键被按下")
14. 57             elif event.key == pygame.K_LEFT:
15. 58                 print("向左方向键被按下")
```

代码2 事件响应函数

代码2第47行代码中，`pygame.event.get()` 的作用是从事件队列获取事件对象，生成待处理事件列表。事件队列中的所有事件对象都将取出来并在事件队列中删除它们。这里把取到的全部事件对象称为“待处理事件列表”。for 循环的作用是遍历待处理事件列表，依次把各个事件对象赋给 `event` 变量。

48~49行用于处理关闭窗口事件，结果是退出程序。

第50行是判别是否收到键盘按下事件。如果是，那么查看事件对象 `event` 的 `key` 属性来判别按下的是哪一个键。

我们先来认识4个按键的 `key` 属性值。以后会接触到更多 `key` 属性值。

1. 下方向键。用常量 `pygame.K_DOWN` 标识。注意要与 `pygame.KEYDOWN` 区别开来。前者是一种 `key` 属性值，后者是一种 `type` 属性值。
2. 上方向键。用常量 `pygame.K_UP` 标识。UP 是向上的意思。
3. 右方向键。用常量 `pygame.K_RIGHT` 标识。RIGHT 是向右的意思。
4. 左方向键。用常量 `pygame.K_LEFT` 标识。LEFT 是向左的意思。

调用 `check_events` 函数

在主循环内调用 `check_events` 函数。如代码3所示。

```
1.  #TetrisGame/main.py
2.  10 def main():
3.  11     #初始化pygame。
4.  12     pygame.init()
5.  ...     #省略部分代码
6.  21
7.  22     #游戏主循环
8.  23     while True:
9.  24         #监视键盘和鼠标事件
10. 25         check_events()
11. ...         #省略绘制屏幕的代码
```

代码3 调用 `check_events` 函数

以上子步骤都完成后，运行程序，按下上、下、左、右方向键看看，正确的话会在左下角的运

行输出小窗口中（如图1左下角所示）看到哪一个方向键被按下的报告。

小结

本步骤实现的功能是捕捉键盘按键事件。比如，当捕捉向上方向键被按下，程序就会报告“向上方向键被按下”。这一步骤是为下一步骤打基础的。下一步骤将实现向左、向右或向下移动方块的功能。

要更好地理解键盘按键事件的响应处理，你要想象到程序运行流程是在反复不断地执行程序主循环（即 main 函数内的 while 循环）。

第02课《显示游戏窗口》中详细描述了程序运行流程。程序主循环包含以下子步骤：

1. 从事件队列取出现有事件对象（包括键盘按键事件），逐一作出响应处理。此处有个 for 循环：

```
1.         for event in pygame.event.get():
2.             ..... #对事件event作出处理
```

2. 绘制游戏窗口元素（含文字、图形、图片等）。

3. 刷新游戏窗口。

完成本步骤全部功能的代码可以从 [Github](#) 下载。

再次提醒，要多尝试，多试错，吃透代码。