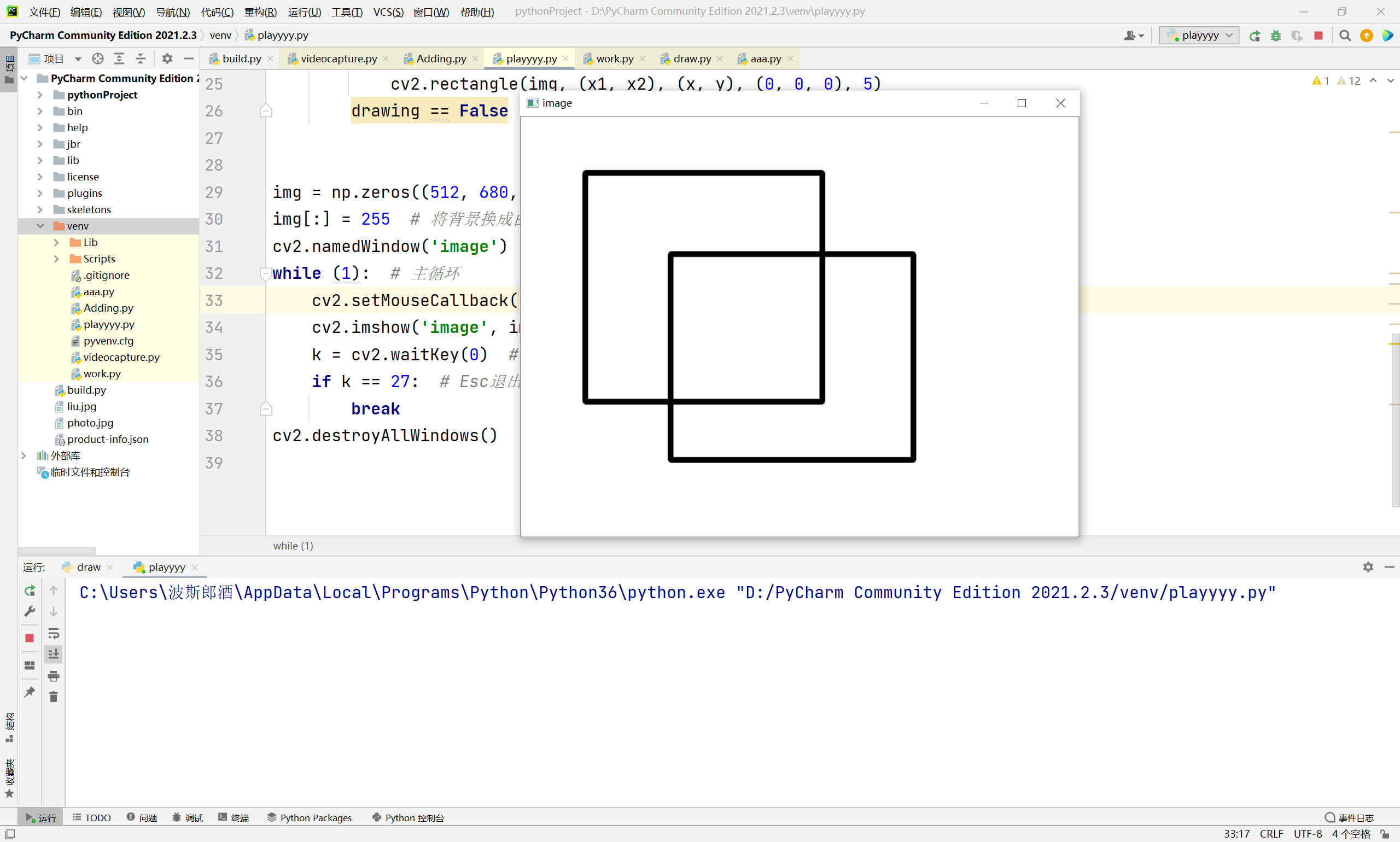


**import** numpy **as** np  
**import** cv2  
  
  
**def** nothing(x):  
 **pass**drawing = **False** *# 鼠标按下变为Ture*x1, x2 = -1, -1 *# 存放画矩形时的左上角坐标*mode = **True** *# 标记左键是否切换***def** draw(event, x, y, flags, param): *# 创建回调函数* **global** x1, x2, drawing, mode  
 r = cv2.getTrackbarPos(**'R'**, **'image'**) *# 获得三个滑动条* g = cv2.getTrackbarPos(**'G'**, **'image'**)  
 b = cv2.getTrackbarPos(**'B'**, **'image'**)  
 color = (b, g, r) *# 用三个滑动条数值所对应颜色获取最终颜色* **if** event == cv2.EVENT\_LBUTTONDOWN: *# 按下左键保留起始坐标* drawing = **True** x1, x2 = x, y  
 **elif** event == cv2.EVENT\_MOUSEMOVE **and** flags == cv2.EVENT\_FLAG\_LBUTTON: *# 按下左键并移动鼠标画填充矩形* **if** drawing == **True**:  
 **if** mode == **True**:  
 cv2.rectangle(img, (x1, x2), (x, y), color, -1)  
 **elif** event == cv2.EVENT\_LBUTTONUP **and** mode == **False**: *# 按下2并松开左键获取不填充矩形* cv2.rectangle(img, (x1, x2), (x, y), color, 5)  
 drawing == **False  
 elif** event == cv2.EVENT\_MOUSEMOVE **and** flags == cv2.EVENT\_FLAG\_RBUTTON: *# 按下右键并移动鼠标画圆* **if** drawing == **True**:  
 cv2.circle(img, (x, y), 3, color, -1)  
  
  
img = np.zeros((512, 680, 3), np.uint8)  
img[:] = 255 *# 修改为白底背景*cv2.namedWindow(**'image'**) *# 定义窗口名*cv2.createTrackbar(**'R'**, **'image'**, 0, 255, nothing)  
cv2.createTrackbar(**'G'**, **'image'**, 0, 255, nothing)  
cv2.createTrackbar(**'B'**, **'image'**, 0, 255, nothing)  
cv2.setMouseCallback(**'image'**, draw)  
**while** (1): *# 主循环* cv2.imshow(**'image'**, img) *# 打开image窗口* k = cv2.waitKey(1) & 0xFF *# 延迟时间并等待输入* **if** k == ord(**'1'**):*#1画填充矩形* mode == **True  
 elif** k == ord(**'2'**):*#2画空心矩形* mode = **False  
 elif** k == 27:*#Esc退出* **break**cv2.destroyAllWindows()

附加题



**import** numpy **as** np  
**import** cv2  
  
  
**def** nothing(x):  
 **pass**drawing = **False** *# 鼠标按下变为Ture*x1, x2 = -1, -1 *# 存放画矩形时的左上角坐标***def** draw(event, x, y, flags, param): *# 创建回调函数* **global** x1, x2, drawing  
 **if** event == cv2.EVENT\_LBUTTONDOWN: *# 按下左键保留起始坐标* drawing = **True** x1, x2 = x, y  
 **if** event == cv2.EVENT\_MOUSEMOVE **and** flags == cv2.EVENT\_FLAG\_LBUTTON: *# 按下左键并移动鼠标* pd = img.copy()*#将pd中图像显示到img中* **if** drawing == **True**:  
 cv2.rectangle(pd, (x1, x2), (x, y), (0, 0, 0), 5)  
 cv2.imshow(**'image'**, pd)*#将pd中图像显示到image窗口* **elif** event == cv2.EVENT\_LBUTTONUP: *# 松开鼠标停止绘画* **if** drawing == **True**:  
 cv2.rectangle(img, (x1, x2), (x, y), (0, 0, 0), 5)  
 drawing == **False**img = np.zeros((512, 680, 3), np.uint8)  
img[:] = 255 *# 将背景换成白色*cv2.namedWindow(**'image'**) *# 定义窗口名***while** (1): *# 主循环* cv2.setMouseCallback(**'image'**, draw)  
 cv2.imshow(**'image'**, img) *# 打开image窗口显示img中内容* k = cv2.waitKey(0) *# 延迟时间并等待输入* **if** k == 27: *# Esc退出* **break**cv2.destroyAllWindows()