

MouseEvent

W4

MouseEvent class

ใน open cv จะมี class ของการทำงานกับเม้าส์ โดยหลักการทำงานมีขั้นตอนการทำงานต่อไปนี้

- 1.กำหนดการทำงานกับเม้าส์ ด้วยการใช้คำสั่ง **SetMouseCallback()** โดยกำหนด ได้ดังนี้
 - 1.การกำหนดชื่อหน้าต่าง ซึ่งต้องเป็น หน้าต่างเดียวกับหน้าต่าง **CV2.imshow()** ที่กำหนดไว้
 - 2.ตามด้วยฟังก์ชันการทำงาน
- 2.การกำหนดฟังก์ชันโดยการผ่าน **Parameter event ตำแหน่ง x และ y , flags ,param**
- 3.การกำหนดเงื่อนไข ที่ตรงกับเหตุการณ์ Event ที่ใช้ในการกดเม้าส์

ຕັ້ງອຢ່າງ



```
1 import cv2 as cv
2 img = cv.imread("image/tree-736885_1280.jpg")
3
4 def click_position(event,x,y,flages,param):
5     if event == cv.EVENT_LBUTTONDOWN:
6         cv.putText(img, "tanapoom", (x,y),1,3,(0,0,255),3)
7         cv.imshow("click puttext",img)
8
9 cv.imshow("click puttext",img)
10
11 cv.setMouseCallback("click puttext",click_position)
12 cv.waitKey(0)
13 cv.destroyAllWindows()
```

ตัวอย่าง

```
● ● ●  
1 import cv2 as cv  
2 img = cv.imread("image/tree-736885_1280.jpg")  
3  
4 def click_position(event,x,y,flages,param):  
5     if event == cv.EVENT_LBUTTONDOWN:  
6         cv.putText(img, "tanapoom", (x,y),1,3,(0,0,255),3)  
7         cv.imshow("click puttext",img)  
8  
9 cv.imshow("click puttext",img)  
10  
11 cv.setMouseCallback("click puttext",click_position)  
12 cv.waitKey(0)  
13 cv.destroyAllWindows()
```

กำหนด **cv2.setMouseCallback()** โดย
ชื่อ หน้าต่าง ต้องตรงกับ
cv2.imshow()ตามด้วยพังก์ชัน

ตัวอย่าง

```
● ● ●  
1 import cv2 as cv  
2 img = cv.imread("image/tree-736885_1280.jpg")  
3  
4 def click_position(event,x,y,flages,param):  
5     if event == cv.EVENT_LBUTTONDOWN:  
6         cv.putText(img, "tanapoom", (x,y),1,3,(0,0,255),3)  
7         cv.imshow("click puttext",img)  
8  
9 cv.imshow("click puttext",img)  
10  
11 cv.setMouseCallback("click puttext",click_position)  
12 cv.waitKey(0)  
13 cv.destroyAllWindows()
```

ฟังก์ชัน **click_position**
นี้จะต้องผ่านค่าต่างๆ ดังนี้

1.event คือเหตุการณ์ที่ กำ
หนด เช่น คลิกเม้าส์ซ้าย และ
ขวา

2. X และ y คือตำแหน่งของ
จัตุรัสใน **putText**

3. Flags คือรูปแบบ **font**
linetype

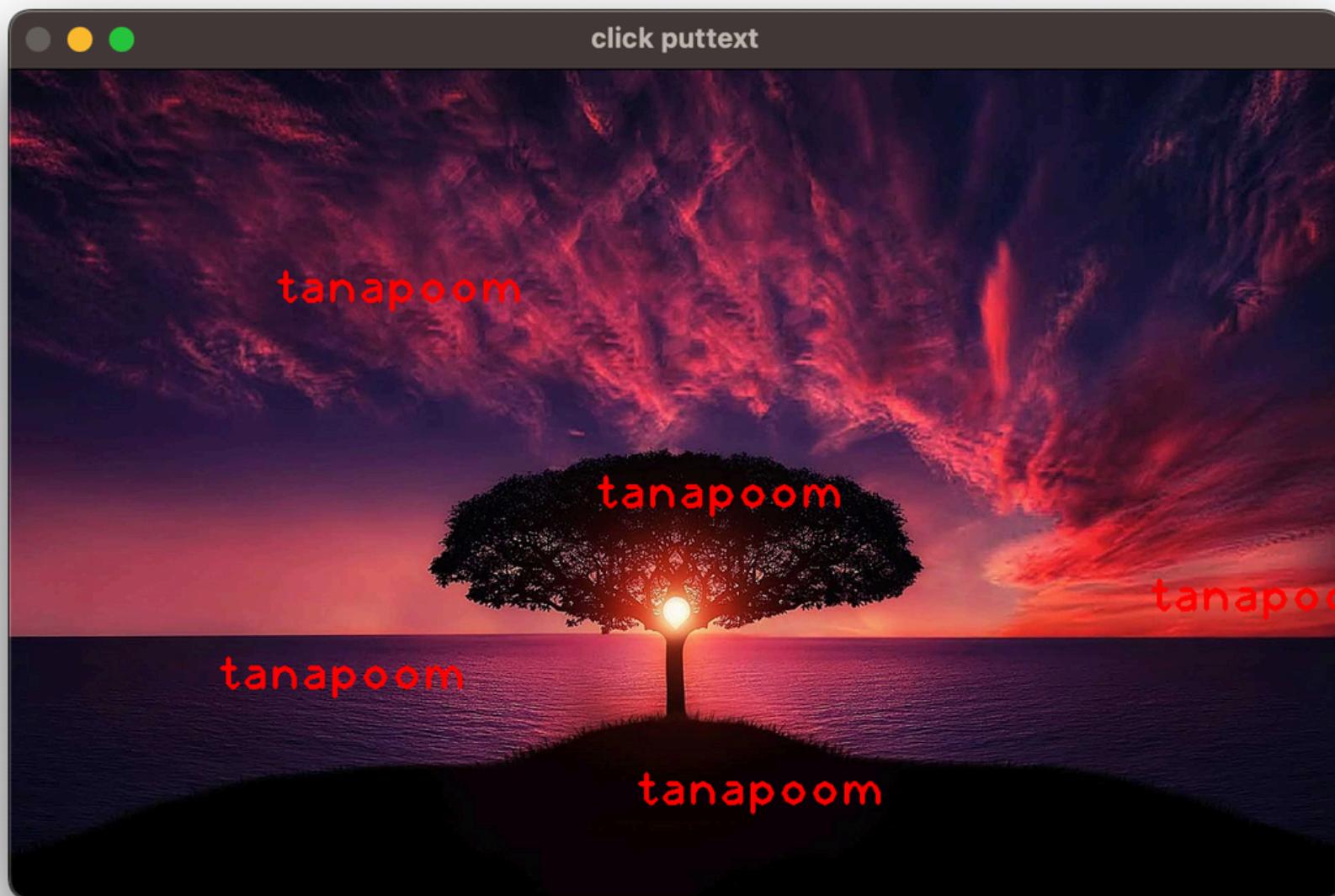
4. Param คือ
parameter ที่ต้องใส่ในฟัง
ก์ชัน เช่น สี ความหนา

ตัวอย่าง

```
● ● ●  
1 import cv2 as cv  
2 img = cv.imread("image/tree-736885_1280.jpg")  
3  
4 def click_position(event,x,y,flages,param):  
5     if event == cv.EVENT_LBUTTONDOWN:  
6         cv.putText(img, "tanapoom", (x,y),1,3,(0,0,255),3)  
7         cv.imshow("click puttext",img)  
8  
9 cv.imshow("click puttext",img)  
10  
11 cv.setMouseCallback("click puttext",click_position)  
12 cv.waitKey(0)  
13 cv.destroyAllWindows()
```

event การกดเมาส์ โดย
คำสั่งนี้เป็นการตรวจสอบ
การกดเมาส์

ຕັ້ວອຍ່າງ



การประยุกต์ใช้งาน

MouseEvent เพื่อบอกตำแหน่ง

ใน open cv จะมี class ของการทำงานกับเม้าส์ โดยหลักการทำงานมีขั้นตอนการทำงานต่อไปนี้

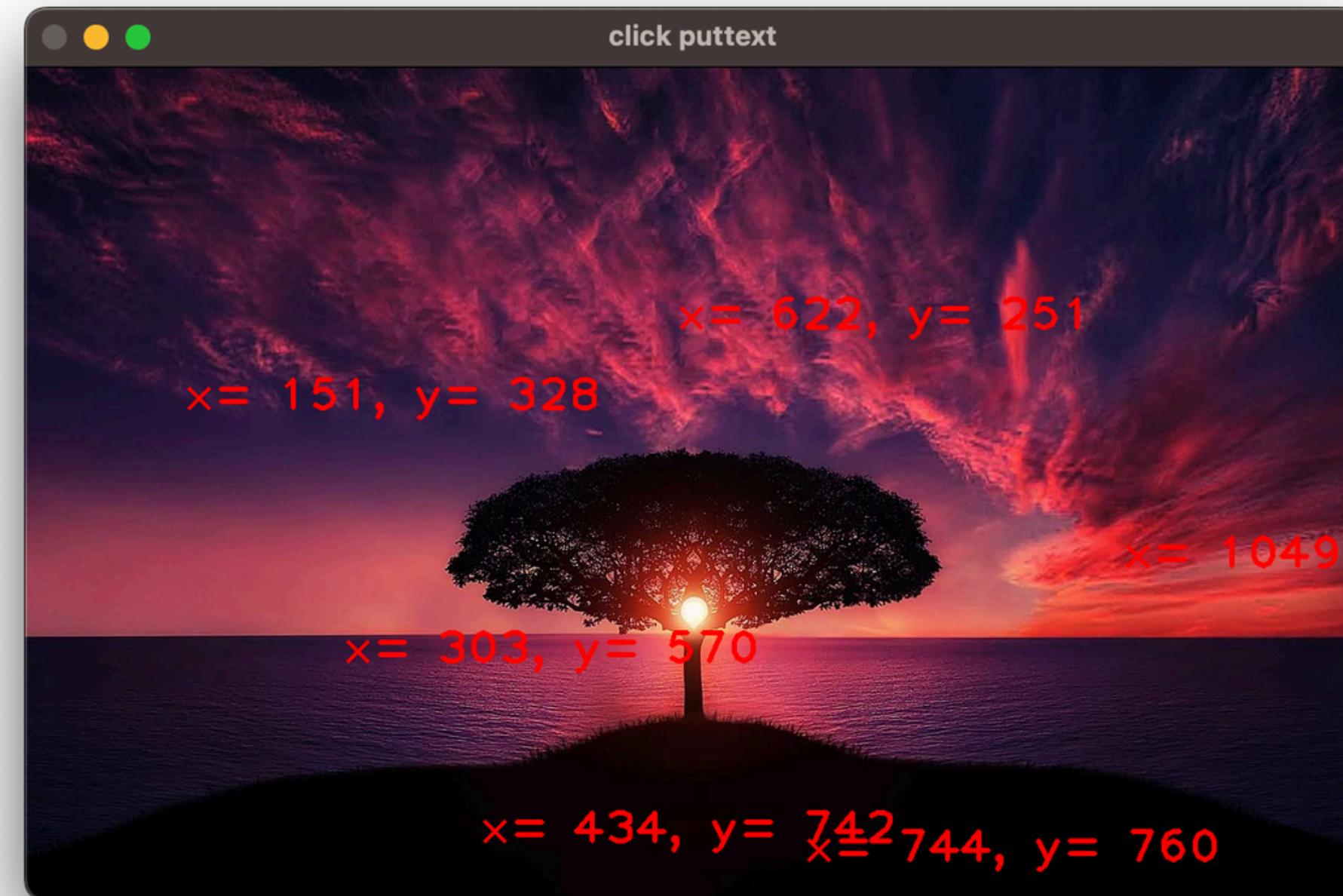
- 1.กำหนดการทำงานกับเม้าส์ ด้วยการใช้คำสั่ง `SetMouseCallback()` โดยกำหนด ได้ดังนี้
 - 1.การกำหนดชื่อหน้าต่าง ซึ่งต้องเป็น หน้าต่างเดียวกับหน้าต่าง `CV2.imshow()` ที่กำหนดไว้
 - 2.ตามด้วยฟังก์ชันการทำงาน
- 2.การกำหนดฟังก์ชันโดยการผ่าน Parameter event ตำแหน่ง x และ y , flags ,param
- 3.การกำหนดเงื่อนไข ที่ตรงกับเหตุการณ์ Event ที่ใช้ในการกดเม้าส์
- 4.กำหนดตัวแปร text ให้เท่ากับ x,y

MouseEvent เพื่อบอกตำแหน่ง



```
1 def click_position(event,x,y,flages,param):  
2     if event == cv.EVENT_LBUTTONDOWN:  
3         text = ("x= %s, y= %s"%(x,y))  
4         cv.putText(img, text,(x,y),1,3,(0,0,255),3)  
5         cv.imshow("click puttext",img)
```

MouseEvent เพื่อบอกตำแหน่ง



MouseEvent เพื่อแสดงค่าสี

ใน open cv จะมี class ของการทำงานกับเม้าส์ โดยหลักการทำงานมีขั้นตอนการทำงานต่อไปนี้

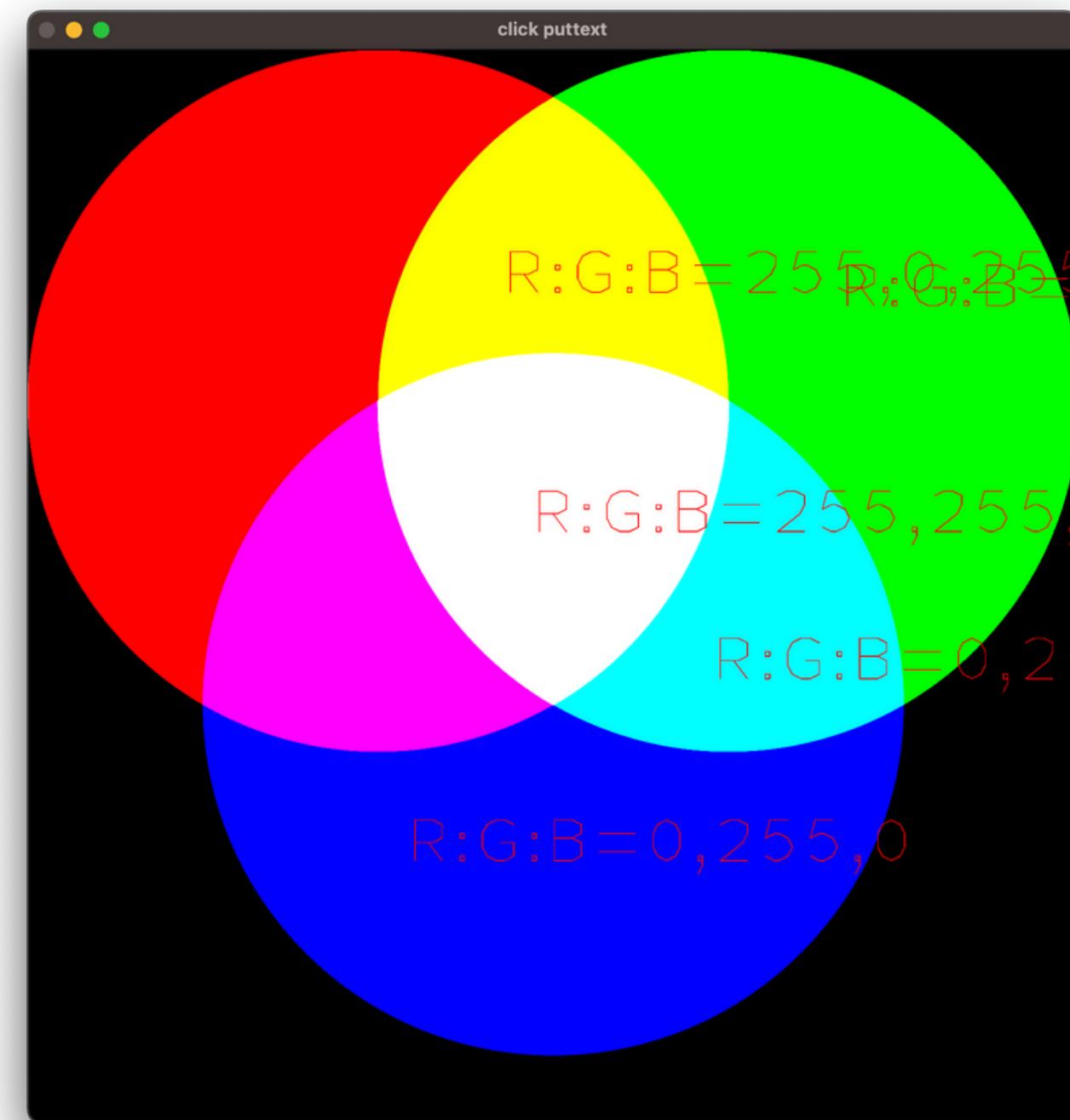
- 1.กำหนดการทำงานกับเม้าส์ ด้วยการใช้คำสั่ง `SetMouseCallback()` โดยกำหนด ได้ดังนี้
 - 1.การกำหนดชื่อหน้าต่าง ซึ่งต้องเป็น หน้าต่างเดียวกับหน้าต่าง `CV2.imshow()` ที่กำหนดไว้
 - 2.ตามด้วยฟังก์ชันการทำงาน
- 2.การกำหนดฟังก์ชันโดยการผ่าน Parameter event ตำแหน่ง x และ y , flags ,param
- 3.การกำหนดเงื่อนไข ที่ตรงกับเหตุการณ์ Event ที่ใช้ในการกดเม้าส์
- 4.กำหนดตัวแปร text ให้เท่ากับ red,green,blue

MouseEvent เพื่อแสดงค่าสี



```
1 def click_position(event,x,y,flages,param):
2     if event == cv.EVENT_LBUTTONDOWN:
3         blue = img[x,y,0]
4         green = img[x,y,1]
5         red = img[x,y,2]
6         text = ("R:G:B=%s,%s,%s"%(red,green,blue))
7         cv.putText(img, text,(x,y),1,3,(0,0,255),3)
8         cv.imshow("click puttext",img)
```

MouseEvent เพื่อแสดงค่าสี

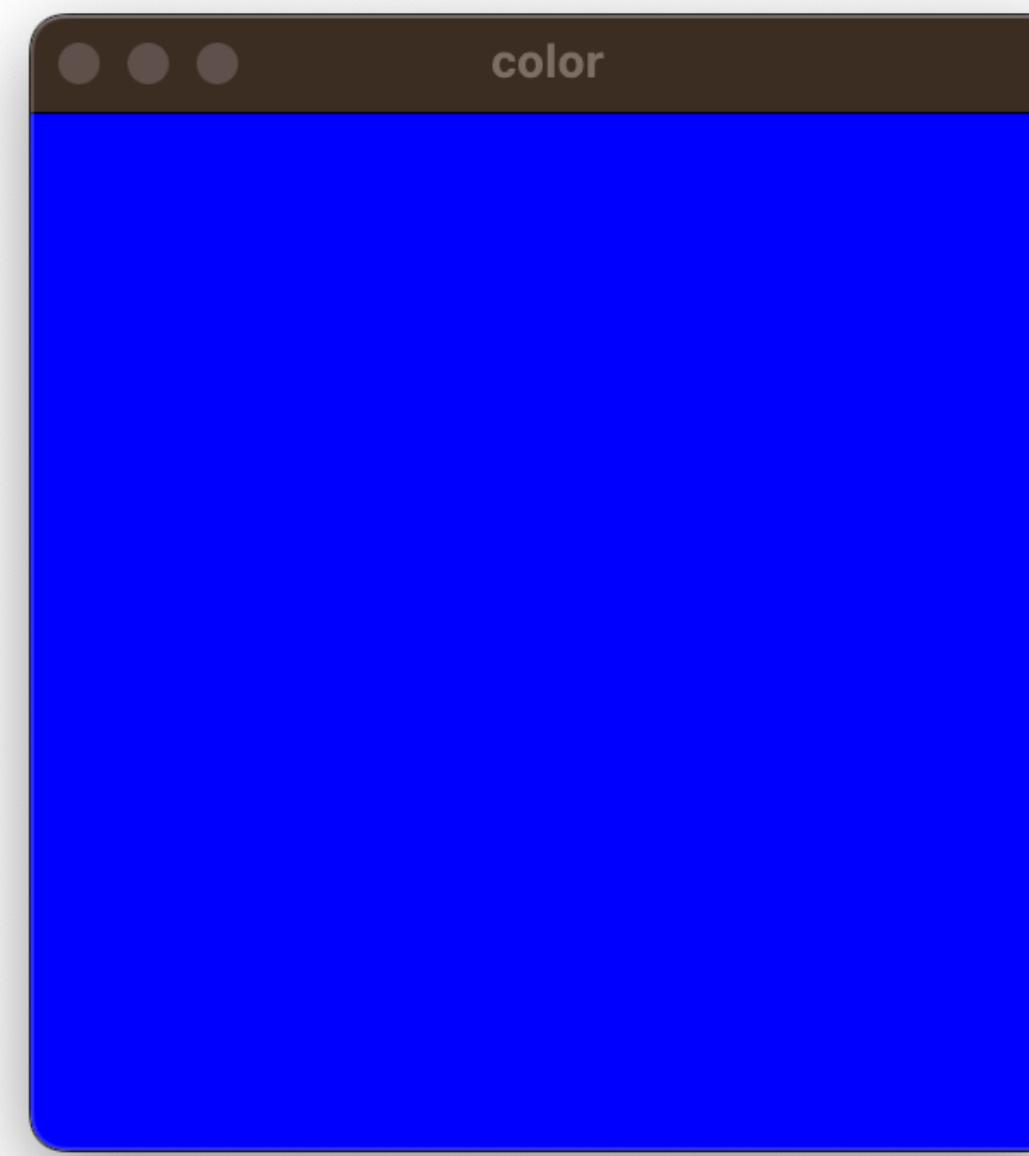
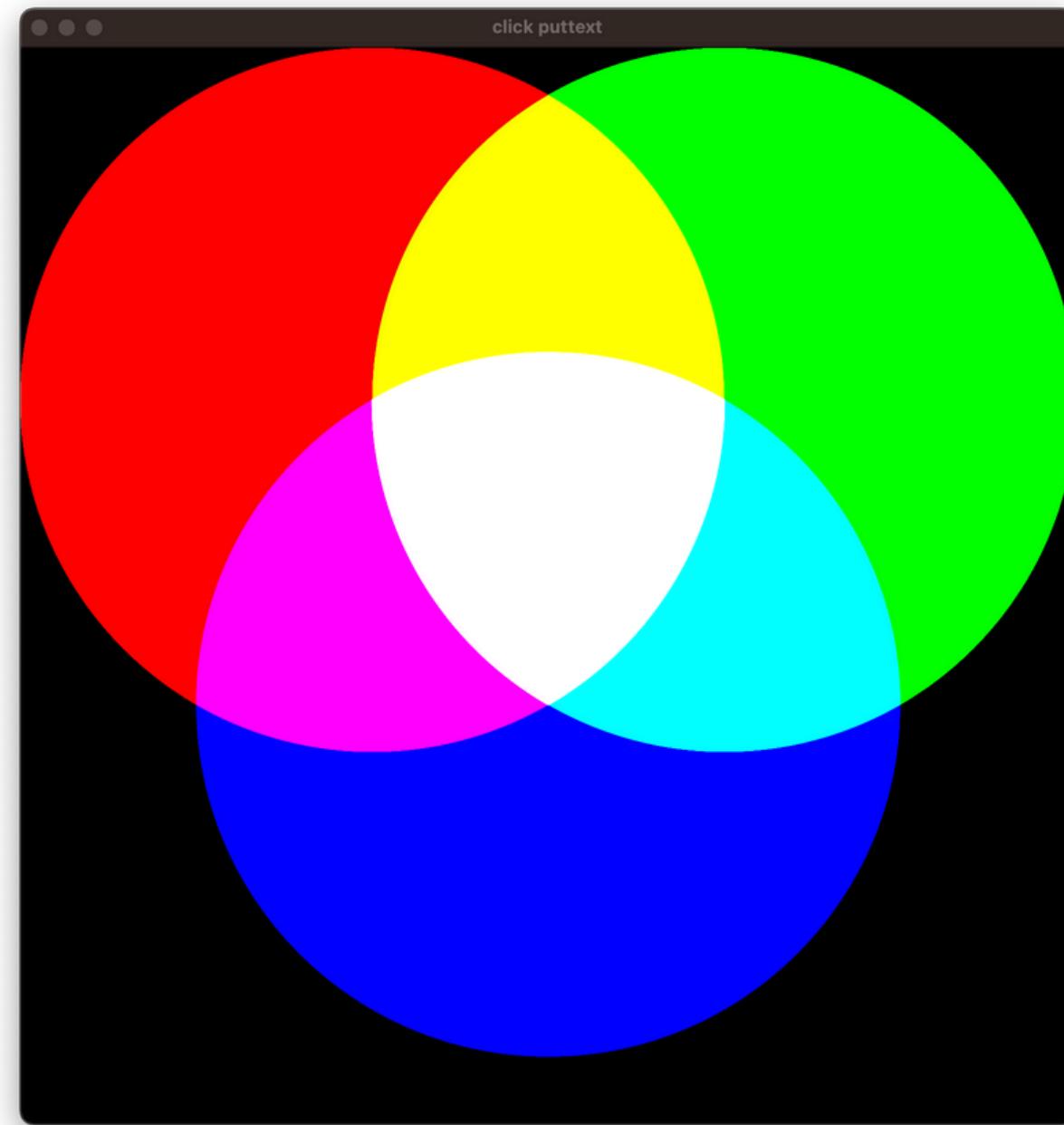


MouseEvent เพื่อแสดงรูปสี



```
1 def click_position(event,x,y,flages,param):
2     if event == cv.EVENT_LBUTTONDOWN:
3         blue = img[x,y,0]
4         green = img[x,y,1]
5         red = img[x,y,2]
6         imgcolor = np.zeros([300,300,3],np.uint8)
7         imgcolor[:] = [green,blue,red]
8         cv.imshow("color",imgcolor)
```

MouseEvent เพื่อแสดงรูปสี



MouseEvent เพื่อบอกสร้างเส้นต่อเนื่อง

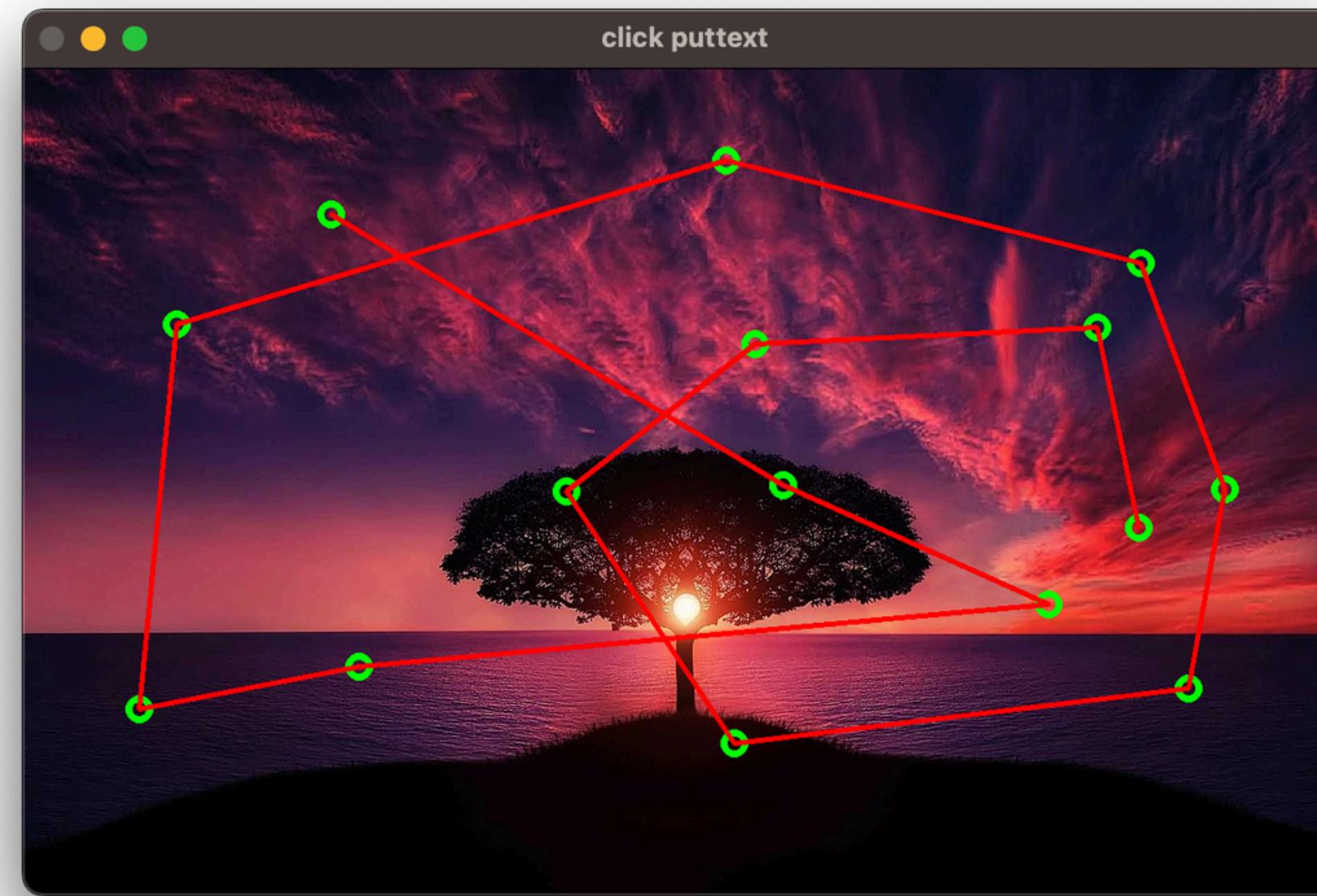
ใน open cv จะมี class ของการทำงานกับเม้าส์ โดยหลักการทำงานมีขั้นตอนการทำงานต่อไปนี้

- 1.กำหนดการทำงานกับเม้าส์ ด้วยการใช้คำสั่ง `SetMouseCallback()` โดยกำหนด ได้ดังนี้
 - 1.การกำหนดชื่อหน้าต่าง ซึ่งต้องเป็น หน้าต่างเดียวกับหน้าต่าง `CV2.imshow()` ที่กำหนดไว้
 - 2.ตามด้วยฟังก์ชันการทำงาน
- 2.การกำหนดฟังก์ชันโดยการผ่าน Parameter event ตำแหน่ง x และ y , flags ,param
- 3.การกำหนดเงื่อนไข ที่ตรงกับเหตุการณ์ Event ที่ใช้ในการกดเม้าส์
- 4.การกำหนดตัวแปร point ให้เก่ากับรายการว่างเปล่า [] และให้สร้างวงกลม `cv.circle()` ตามตำแหน่งของ การ click mouse บุ้ม ซ้าย และนำค่า x,y ไปเพิ่ม ในตัวแปร point

MouseEvent เพื่อบอกสร้างเส้นต่อเนื่อง

```
● ● ●  
1 point = []  
2 def click_position(event,x,y,flages,param):  
3     if event == cv.EVENT_LBUTTONDOWN:  
4         cv.circle(img,(x,y),10,(0,255,0),5)  
5         point.append((x,y))  
6         if len(point) >= 2:  
7             cv.line(img,point[-1],point[-2],(0,0,255),3)  
8         cv.imshow("click puttext",img)
```

MouseEvent เพื่อบอกสร้างเส้นต่อเนื่อง



แบบฝึกหัด

ให้นักเรียนสร้างโปรแกรมว่าด้วยรูป 4
เหลี่ยมจากการลากเม้าส์

