## **Trackbar เป็น Palette สี**

เป้าหมาย

## เรียนรู้การผูก trackbar กับหน้าต่าง OpenCV

* คุณจะได้เรียนรู้ฟังก์ชันเหล่านี้: **cv2.getTrackbarPos ()** , **cv2.createTrackbar ()**เป็นต้น

CODE Demo

## ที่นี่เราจะสร้างแอปพลิเคชันง่ายๆซึ่งจะแสดงสีที่คุณระบุ คุณมีหน้าต่างที่แสดงสีและแถบแทร็กสามแท่งเพื่อระบุแต่ละสี B, G, R คุณเลื่อนแถบแทร็คและการเปลี่ยนแปลงสีของหน้าต่าง สีเริ่มต้นจะถูกตั้งเป็นสีดำ

สำหรับฟังก์ชัน **cv2.getTrackbarPos ()** อาร์กิวเมนต์แรกคือชื่อแทร็คที่สองคือชื่อหน้าต่างที่มีการแนบอาร์กิวเมนต์ที่สามคือค่าดีฟอลต์หนึ่งในสี่คือค่าสูงสุดและหนึ่งในห้าคือฟังก์ชันเรียกกลับที่ทำงาน การเปลี่ยนแปลงค่าแทร็คบาร์ทุกครั้ง ฟังก์ชัน **callback** มีอาร์กิวเมนต์เริ่มต้นซึ่งเป็นตำแหน่ง **trackbar** อยู่เสมอ ในกรณีของเราฟังก์ชันไม่ทำอะไรเลยเราจึงผ่านไปได้

แอ็พพลิเคชันที่สำคัญอีกอย่างหนึ่งของ **trackbar** คือการใช้เป็นปุ่มหรือสวิทช์ OpenCV โดยค่าเริ่มต้นไม่มีฟังก์ชันปุ่ม เพื่อให้คุณสามารถใช้ **trackbar** เพื่อให้ได้ฟังก์ชันดังกล่าว ในแอ็พพลิเคชันของเราเราได้สร้างสวิทช์หนึ่งตัวในแอพพลิเคชันที่ใช้งานได้เฉพาะเมื่อเปิดสวิตช์มิฉะนั้นหน้าจอจะเป็นสีดำเสมอ

**import** **cv2**

**import** **numpy** **as** **np**

**def** nothing(x):

**pass**

*# Create a black image, a window*

img = np.zeros((300,512,3), np.uint8)

cv2.namedWindow('image')

*# create trackbars for color change*

cv2.createTrackbar('R','image',0,255,nothing)

cv2.createTrackbar('G','image',0,255,nothing)

cv2.createTrackbar('B','image',0,255,nothing)

*# create switch for ON/OFF functionality*

switch = '0 : OFF **\n**1 : ON'

cv2.createTrackbar(switch, 'image',0,1,nothing)

**while**(1):

cv2.imshow('image',img)

k = cv2.waitKey(1) & 0xFF

**if** k == 27:

**break**

*# get current positions of four trackbars*

r = cv2.getTrackbarPos('R','image')

g = cv2.getTrackbarPos('G','image')

b = cv2.getTrackbarPos('B','image')

s = cv2.getTrackbarPos(switch,'image')

**if** s == 0:

img[:] = 0

**else**:

img[:] = [b,g,r]

cv2.destroyAllWindows()

ภาพหน้าจอของแอปพลิเคชันมีลักษณะดังนี้:

