

# 03-Direct-Drawing-with-Mouse-Set\_The\_Third

August 26, 2025

## 1 Direct Drawing with Mouse

1.1 NOTE: THESE SHOULD ALL BE RUN AS A .py SCRIPT.

1.2 SCRIPT 1: Connecting a Function for Drawing

```
[ ]: import cv2
import numpy as np
# Create a function based on a CV2 Event (Left button click)
def draw_circle(event,x,y,flags,param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(img,(x,y),100,(0,255,0),-1)

# Create a black image
img = np.zeros((512,512,3), np.uint8)
# This names the window so we can reference it
cv2.namedWindow(winname='my_drawing')
# Connects the mouse button to our callback function
cv2.setMouseCallback('my_drawing',draw_circle)

while True: #Runs forever until we break with Esc key on keyboard
    # Shows the image window
    cv2.imshow('my_drawing',img)
    # EXPLANATION FOR THIS LINE OF CODE:
    # https://stackoverflow.com/questions/35372700/whats-0xff-for-in-cv2-waitkey1/39201163
    ↪ if cv2.waitKey(20) & 0xFF == 27:
        break
# Once script is done, its usually good practice to call this line
# It closes all windows (just in case you have multiple windows called)
cv2.destroyAllWindows()
```

1.3 SCRIPT 2: Adding Functionality with Event Choices

---

```
[ ]: import cv2
import numpy as np
```

```

# Create a function based on a CV2 Event (Left button click)
def draw_circle(event,x,y,flags,param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(img,(x,y),100,(0,255,0),-1)
    elif event == cv2.EVENT_RBUTTONDOWN:
        cv2.circle(img,(x,y),100,(0,0,255),-1)

# Create a black image
img = np.zeros((512,512,3), np.uint8)
# This names the window so we can reference it
cv2.namedWindow(winname='my_drawing')
# Connects the mouse button to our callback function
cv2.setMouseCallback('my_drawing',draw_circle)

while True: #Runs forever until we break with Esc key on keyboard
    # Shows the image window
    cv2.imshow('my_drawing',img)
    # EXPLANATION FOR THIS LINE OF CODE:
    # https://stackoverflow.com/questions/35372700/whats-0xff-for-in-cv2-waitkey1/39201163
    if cv2.waitKey(20) & 0xFF == 27:
        break
# Once script is done, its usually good practice to call this line
# It closes all windows (just in case you have multiple windows called)
cv2.destroyAllWindows()

```

## 2 SCRIPT 3: Dragging with Mouse

```

[ ]: import cv2
import numpy as np

# Create a function based on a CV2 Event (Left button click)

# mouse callback function
def draw_rectangle():

```

```

# Connects the mouse button to our callback function
cv2.setMouseCallback('my_drawing',draw_rectangle)

while True: #Runs forever until we break with Esc key on keyboard
    # Shows the image window
    cv2.imshow('my_drawing',img)
    # EXPLANATION FOR THIS LINE OF CODE:
    # https://stackoverflow.com/questions/35372700/whats-0xff-for-in-cv2-waitkey1/39201163

    # CHECK TO SEE IF ESC WAS PRESSED ON KEYBOARD
    if cv2.waitKey(1) & 0xFF == 27:
        break
# Once script is done, its usually good practice to call this line
# It closes all windows (just in case you have multiple windows called)
cv2.destroyAllWindows()

```

[ ]: