



Mouse callback function

We can access the mouse inside a window by using `cv2.setMouseCallback()`

Creating mouse callback function has a specific format. It differs only in what the function does.

The format of arguments in the callback function is as follows

`(event,x,y,flags,param)`

where

- x,y ---> are the current mouse position on the canvas
- Event ---> mouse event trigger captured by the opencv
- flags ---> Is the additional triggers passed by the opencv (eg keyboard click along with the main mouse event)
- params ---> Is the parameters that can be passed by the user while calling the function

Use the following code to get a list of all the EVENTS and FLAGS

In [1]:

```
import cv2
events = [i for i in dir(cv2) if 'EVENT' in i]
print(events)
```

```
['EVENT_FLAG_ALTKEY', 'EVENT_FLAG_CTRLKEY', 'EVENT_FLAG_LBUTTON', 'EVENT_FLAG_MBUTTON',
 'EVENT_FLAG_RBUTTON', 'EVENT_FLAG_SHIFTKEY', 'EVENT_LBUTTONDBLCLK', 'EVENT_LBUTTONDOWN',
 'EVENT_LBUTTONUP', 'EVENT_MBUTTONDBLCLK', 'EVENT_MBUTTONDOWN', 'EVENT_MBUTTONUP', 'EVENT_
_MOUSEHWHEEL', 'EVENT_MOUSEMOVE', 'EVENT_MOUSEWHEEL', 'EVENT_RBUTTONDBLCLK', 'EVENT_RBUT
TONTDOWN', 'EVENT_RBUTTONUP']
```

Practice Problem : Drawing Circles on Double Click

In []:

```
import cv2
import numpy as np

# mouse callback function
def draw_circle(event,x,y,flags,param):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        cv2.circle(img,(x,y),100,(255,0,0),-1)

# Create a black image, a window and bind the function to window
img = np.zeros((512,512,3), np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)

while(1):
    cv2.imshow('image',img)
    if cv2.waitKey(20) & 0xFF == 27:
```

```
        break
cv2.destroyAllWindows()
```

Practice Problem : Drawing Circles in two windows with different color on Double Click

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

def draw_red(event,x,y,flag,param):
    if event ==cv2.EVENT_LBUTTONDBLCLK:
        centre=(100,100)
        cv2.circle(img,(x,y),50,(0,0,255),-5)

def draw_blue(event,x,y,flag,param):
    if event ==cv2.EVENT_LBUTTONDBLCLK: #try with other event
        centre=(200,100)
        cv2.circle(img,(x,y),70,(255,0,0),-5)

img=np.zeros((600,600,3),np.uint8)
cv2.namedWindow('red')
cv2.setMouseCallback('red',draw_red)
cv2.namedWindow('blue')
cv2.setMouseCallback('blue',draw_blue)

while True:
    cv2.imshow('red',img)
    cv2.imshow('blue',img)
    if cv2.waitKey(20)&0xFF==27:
        break
cv2.destroyAllWindows()
```

Practice Problem : Paint Brush Application with move function

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

# mouse callback function
def draw_circle(event,x,y,flags,param):
    if event == cv2.EVENT_MOUSEMOVE:
        cv2.circle(img,(x,y),50,(255,0,0),-1)

# Create a black image, a window and bind the function to window
img = np.zeros((512,512,3), np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)

while(1):
    cv2.imshow('image',img)
    if cv2.waitKey(20) & 0xFF == 27:
        break
cv2.destroyAllWindows()
```

Practice Problem : Paint Brush Application move and click function

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

drawing = False # true if mouse is pressed
mode = True # if True, draw rectangle. Press 'm' to toggle to curve
ix,iy = -1,-1

# mouse callback function
def draw_circle(event,x,y,flags,param):
    global ix,iy,drawing,mode

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        ix,iy = x,y

    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == True:
            if mode == True:
                cv2.rectangle(img,(ix,iy),(x,y),(0,255,0),-1)
            else:
                cv2.circle(img,(x,y),5,(0,0,255),-1)

    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False
        if mode == True:
            cv2.rectangle(img,(ix,iy),(x,y),(0,255,0),-1)
        else:
            cv2.circle(img,(x,y),5,(0,0,255),-1)

img = np.zeros((512,512,3), np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)

while(1):
    cv2.imshow('image',img)
    k = cv2.waitKey(1) & 0xFF
    if k == ord('m'):
        mode = not mode
    elif k == 27:
        break

cv2.destroyAllWindows()
```

In []:

Create an Image cropping tool

```
In [2]: import cv2
import numpy as np

drawing = 0 # true if mouse is pressed
crop = False
x1,y1 = -1,-1
crop_coordinates = []
img_crop = []
```

```

img = cv2.imread("images\\lena.jpg",1)
img_copy = np.copy(img)

# mouse callback function
def draw_circle(event,x,y,flags,param):
    global x1,y1,drawing,mode,crop_coordinates,crop
    global img_copy

    #To check button is pressed to crop the Area
    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = 1
        x1,y1 = x,y

    #To select the cropping area
    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing == 1 and drawing!=2:
            img_copy = np.copy(img)

            cv2.rectangle(img_copy,(x1,y1),(x,y),(0,255,0),2)

    #To collect the cordinates of the cropping area
    elif event == cv2.EVENT_LBUTTONUP:
        if drawing == 1:
            drawing = 0
            cv2.rectangle(img_copy,(x1,y1),(x,y),(0,255,0),2)
            crop_coordinates = [x1,y1,x,y]
            crop = True

cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)

while(1):
    cv2.imshow('image',img_copy)
    k = cv2.waitKey(1) & 0xFF
    if k == ord('c'):
        if crop == True:
            crop = False
            img_crop = img_copy[crop_coordinates[1]:crop_coordinates[3],crop_coordinates[0]:crop_coordinates[2]]
            print(type(img_crop))
            cv2.imshow('preview',img_crop)
        elif k == 27:
            break

cv2.destroyAllWindows()

```

```
<class 'numpy.ndarray'>
```

HOMEWORK

1. Write a program to draw different shapes on different mouse click
2. Write a program to create tic tac toe game with image processing

HOMEWORK SOLUTION

In [12]:

```

#TASK 1

import cv2

```

```

import numpy as np

# mouse callback function
def draw_circle(event,x,y,flags,param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(img,(x,y),100,(255,0,0),-1)

    if event == cv2.EVENT_RBUTTONDOWN:
        cv2.rectangle(img,(x,y),(510,128),(0,255,0),3)

    # add more function
img = np.zeros((512,512,3), np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)

while(1):
    cv2.imshow('image',img)
    if cv2.waitKey(20) & 0xFF == 27:
        break
cv2.destroyAllWindows()

```

In []:

```

#TASK 2
import numpy as np
from cv2 import cv2
import Lesson_7_2

img = np.zeros((600,600,3),dtype = 'uint8')

B = [" ", " ", " ", " ", " ", " ", " ", " ", " ", " "]
count=0

cv2.line(img,(200,0),(200,600),(255,255,255),2)
cv2.line(img,(400,0),(400,600),(255,255,255),2)
cv2.line(img,(0,200),(600,200),(255,255,255),2)
cv2.line(img,(0,400),(600,400),(255,255,255),2)

def winner(result):

    if result=="x won":
        cv2.putText(img,'X WON',(300,300),cv2.FONT_HERSHEY_PLAIN,5,(255,255,255),2)
    elif result=="o won":
        cv2.putText(img,'O WON',(300,300),cv2.FONT_HERSHEY_PLAIN,5,(255,255,255),2)

def win_check(B):
    if B[0]=='X' and B[1] == 'X' and B[2] == 'X':
        return "x won"
    elif B[0] == 'X' and B[4] == 'X' and B[8] == 'X':
        return "x won"
    elif B[0] == 'X' and B[3] == 'X' and B[6] == 'X':
        return "x won"
    elif B[2] == 'X' and B[4] == 'X' and B[6] == 'X':
        return "x won"
    elif B[1] == 'X' and B[4] == 'X' and B[7] == 'X':
        return "x won"
    elif B[2] == 'X' and B[5] == 'X' and B[8] == 'X':
        return "x won"
    elif B[6] == 'X' and B[7] == 'X' and B[8] == 'X':
        return "x won"

```

```

elif B[3] == 'X' and B[4] == 'X' and B[5] == 'X':
    return "x won"

elif B[0] == 'O' and B[1] == 'O' and B[2] == 'O':
    return "o won"
elif B[0] == 'O' and B[4] == 'O' and B[8] == 'O':
    return "o won"
elif B[0] == 'O' and B[3] == 'O' and B[6] == 'O':
    return "o won"
elif B[2] == 'O' and B[4] == 'O' and B[6] == 'O':
    return "o won"
elif B[1] == 'O' and B[4] == 'O' and B[7] == 'O':
    return "o won"
elif B[2] == 'O' and B[5] == 'O' and B[8] == 'O':
    return "o won"
elif B[6] == 'O' and B[7] == 'O' and B[8] == 'O':
    return "o won"
elif B[3] == 'O' and B[4] == 'O' and B[5] == 'O':
    return "o won"
print(B)

def something(event,x,y,flag,params):
    global count, B
    if event == cv2.EVENT_LBUTTONDOWN:
        if x<200 and y<200 and B[0]==" ":
            if count%2==0:
                cv2.line(img,(0,0),(200,200),(255,255,255),2)
                cv2.line(img,(200,0),(0,200),(255,255,255),2)
                B[0]="X"
            else:
                cv2.circle(img,(100,100),100,(255,255,255),2)
                B[0]="O"

            result=Lesson_7_2.win_check(B)
            winner(result)

            count+=1

        elif 200<x<400 and y<200 and B[1]==" ":
            if count%2==0:
                cv2.line(img,(200,200),(400,0),(255,255,255),2)
                cv2.line(img,(200,0),(400,200),(255,255,255),2)
                B[1]="X"
            else:
                cv2.circle(img,(300,100),100,(255,255,255),2)
                B[1]="O"

            result=Lesson_7_2.win_check(B)
            winner(result)

            count+=1

        elif 400<x<600 and y<200 and B[2]==" ":
            if count%2==0:
                cv2.line(img,(400,200),(600,0),(255,255,255),2)
                cv2.line(img,(400,0),(600,200),(255,255,255),2)
                B[2]="X"
            else:
                cv2.circle(img,(500,100),100,(255,255,255),2)
                B[2]="O"

```

```

result=Lesson_7_2.win_check(B)
winner(result)

count+=1

elif x<200 and 200<y<400 and B[3]==" ":
    if count%2==0:
        cv2.line(img,(0,200),(200,400),(255,255,255),2)
        cv2.line(img,(200,200),(0,400),(255,255,255),2)
        B[3]="X"
    else:
        cv2.circle(img,(100,300),100,(255,255,255),2)
        B[3]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

elif 200<x<400 and 200<y<400 and B[4]==" ":
    if count%2==0:
        cv2.line(img,(200,200),(400,400),(255,255,255),2)
        cv2.line(img,(400,200),(200,400),(255,255,255),2)
        B[4]="X"
    else:
        cv2.circle(img,(300,300),100,(255,255,255),2)
        B[4]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

elif 400<x<600 and 200<y<400 and B[5]==" ":
    if count%2==0:
        cv2.line(img,(400,200),(600,400),(255,255,255),2)
        cv2.line(img,(400,400),(600,200),(255,255,255),2)
        B[5]="X"
    else:
        cv2.circle(img,(500,300),100,(255,255,255),2)
        B[5]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

elif x<200 and 400<y<600 and B[6]==" ":
    if count%2==0:
        cv2.line(img,(0,400),(200,600),(255,255,255),2)
        cv2.line(img,(200,400),(0,600),(255,255,255),2)
        B[3]="X"
    else:
        cv2.circle(img,(100,500),100,(255,255,255),2)
        B[3]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

```

```

elif 200<x<400 and 400<y<600 and B[7]==" ":
    if count%2==0:
        cv2.line(img,(200,400),(400,600),(255,255,255),2)
        cv2.line(img,(400,400),(200,600),(255,255,255),2)
        B[4]="X"
    else:
        cv2.circle(img,(300,500),100,(255,255,255),2)
        B[4]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

elif 400<x<600 and 400<y<600 and B[8]==" ":
    if count%2==0:
        cv2.line(img,(400,400),(600,600),(255,255,255),2)
        cv2.line(img,(600,400),(400,600),(255,255,255),2)
        B[5]="X"
    else:
        cv2.circle(img,(500,500),100,(255,255,255),2)
        B[5]="O"

    result=Lesson_7_2.win_check(B)
    winner(result)

    count+=1

cv2.namedWindow('ttt')
cv2.setMouseCallback('ttt',something)

while True:
    cv2.imshow('ttt',img)
    key = cv2.waitKey(50)
    if key == ord('q'):
        break

cv2.destroyAllWindows()

```