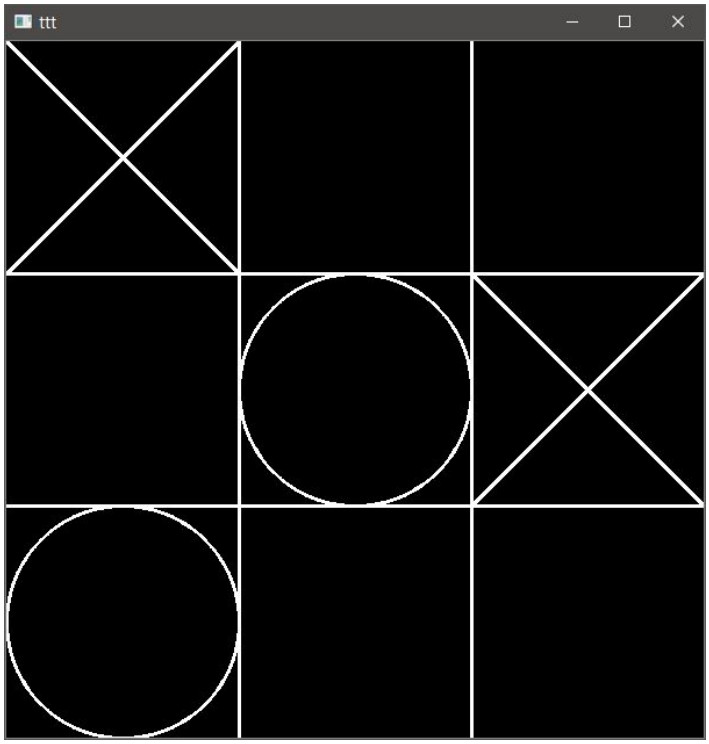
project

The Tic-Tac-Toe Game



In [ ]:

```
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 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()
```

We will using the following module to help us with the game logic

In [ ]:

```
# #   0  |  1  |  2
# # =================
# #   3  |  4  |  5
# # =================
# #   6  |  7  |  8



# B = [" ", " ", " "," ", " ", " "," ", " ", " "]



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)
```

In [ ]: