

Run this with python version 3 or higher:

My GitHub wasn't working, and I tried creating a second account, and it still didn't work there, so I just made a google doc with all of my code.

```
import math
import sys
import numpy as np
import pygame

#variable set up
rownum = int (input("Enter desired height of board:"))
colnum = int(input("Enter desired length of board:"))
over = False
turn = 0

#functions set up
def makeboard():
    board = np.zeros((rownum, colnum))
    return board

def realmove(board, row, column, token):
    board[row][column] = token

def validatenum(board, column):
    return (board[rownum - 1][column] == 0)

def nextrow(board, column):
    for i in range(colnum):
        if board[i][column] == 0:
            return i

def rotateboard(board):
    print(np.flip(board, 0))

def win(board, token):
    #horizontal
    for i in range(colnum - 3):
        for j in range(rownum):
            if board[j][i] == token and board[j][i + 1] == token and board[j][i +
2] == token and board[j][i + 3] == token:
                return True
```

```

#verticle
for i in range(colnum):
    for j in range(rownum - 3):
        if board[j][i] == token and board[j + 1][i] == token and board[j +
2][i] == token and board[j + 3][i] == token:
            return True
#positive diagonal
for i in range(colnum - 3):
    for j in range(rownum - 3):
        if board[j][i] == token and board[j + 1][i + 1] == token and board[j
+ 2][i + 2] == token and board[j + 3][i + 3] == token:
            return True
#negative diagonal
for i in range(colnum - 3):
    for j in range(3, rownum):
        if board[j][i] == token and board[j - 1][i + 1] == token and board[j
- 2][i + 2] == token and board[j - 3][i + 3] == token:
            return True

```

#functions with pygame

```

def pyboard(board):
    for i in range(colnum):
        for j in range(rownum):
            pygame.draw.rect(screen, blue, (i*square, j*square + square, square,
square))
            pygame.draw.circle(screen, black, (int(i*square + square/2) ,
int(j*square + square + square/2)), radius)
    for i in range(colnum):
        for j in range(rownum):
            if board[j][i] == 1:
                pygame.draw.circle(screen, red, (int(i*square + square/2), height-
int(j*square + square/2)), radius)
            elif board[j][i] == 2:
                pygame.draw.circle(screen, yellow, (int(i*square + square/2) ,
height - int(j*square + square/2)), radius)
    pygame.display.update()

```

#starting board setup

```

realboard = makeboard()
print(realboard)

#pygame variable setup
pygame.init()
square = 70
width = colnum * square
height = (rownum + 1)* square
size = (width, height)
screen = pygame.display.set_mode(size)
blue = (25, 108, 254)
black = (0,0,0)
red = (254, 36, 23)
yellow = (238, 248, 0)
radius = int(square/2 - 5)
font = pygame.font.SysFont("impact", 85)

#drawing the board
pyboard(realboard)
pygame.display.update()

#Main Game Loop
while over == False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        if event.type == pygame.MOUSEMOTION:
            pygame.draw.rect(screen, black, (0,0, width, square))
            posx = event.pos[0]
            if turn == 0:
                pygame.draw.circle(screen, red, (posx, int(square/2)), radius)
            else:
                pygame.draw.circle(screen, yellow, (posx, int(square/2)), radius)
    pygame.display.update()
    if event.type == pygame.MOUSEBUTTONDOWN:
        pygame.draw.rect(screen, black, (0,0, width, square))
        #input for player 1 move
        if turn == 0:
            posx = event.pos[0]
            move = int(math.floor(posx/square))

```

```

turn += 1
if validatenum(realboard, move):
    row = nextrow(realboard, move)
    realmove(realboard, row, move, 1)

    if win(realboard, 1):
        label = font.render("Player 1 Wins!", 1, red)
        screen.blit(label, (40,10))
        over = True
    else:
        rotateboard(realboard)

#input for player 2 move
else:
    posx = event.pos[0]
    move = int(math.floor(posx/square))
    turn -= 1
    if validatenum(realboard, move):
        row = nextrow(realboard, move)
        realmove(realboard, row, move, 2)
        if win(realboard, 2):
            label = font.render("Player 2 Wins!", 1, yellow)
            screen.blit(label, (40,10))
            over = True
        else:
            rotateboard(realboard)
pyboard(realboard)

```