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

BLUE = (0, 0, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
YELLOW = (255, 255, 0)

ROW_COUNT = 6
COLUMN_COUNT = 7

def create_board():
    board = np.zeros((ROW_COUNT, COLUMN_COUNT))
    return board
def drop_piece(board, row, col, piece):
    board[row][col] = piece
def is_valid_location(board, col):
    return board[ROW_COUNT - 1][col] == 0
def get_next_open_row(board, col):
    for r in range(ROW_COUNT):
        if board[r][col] == 0:
            return r
def print_board(board):
    print(np.flip(board, 0))
def winning_move(board, piece):

    for c in range(COLUMN_COUNT - 3):
        for r in range(ROW_COUNT):
            if board[r][c] == piece and board[r][c + 1] == piece and
board[r][c + 2] == piece and board[r][
                c + 3] == piece:
                return True

    for c in range(COLUMN_COUNT):
        for r in range(ROW_COUNT - 3):
            if board[r][c] == piece and board[r + 1][c] == piece and
board[r + 2][c] == piece and board[r + 3][
                c] == piece:
                return True

    for c in range(COLUMN_COUNT - 3):
        for r in range(ROW_COUNT - 3):
            if board[r][c] == piece and board[r + 1][c + 1] == piece and
board[r + 2][c + 2] == piece and board[r + 3][
                c + 3] == piece:
                return True

    for c in range(COLUMN_COUNT - 3):
        for r in range(3, ROW_COUNT):
            if board[r][c] == piece and board[r - 1][c + 1] == piece and
board[r - 2][c + 2] == piece and board[r - 3][
                c + 3] == piece:
                return True
def draw_board(board):
    for c in range(COLUMN_COUNT):
        for r in range(ROW_COUNT):
```

```
            pygame.draw.rect(screen, BLUE, (c * SQUARESIZE, r *
SQUARESIZE + SQUARESIZE, SQUARESIZE, SQUARESIZE))
            pygame.draw.circle(screen, BLACK, (
            int(c * SQUARESIZE + SQUARESIZE / 2), int(r * SQUARESIZE +
SQUARESIZE + SQUARESIZE / 2)), RADIUS)

    for c in range(COLUMN_COUNT):
        for r in range(ROW_COUNT):
            if board[r][c] == 1:
                pygame.draw.circle(screen, RED, (
                int(c * SQUARESIZE + SQUARESIZE / 2), height - int(r *
SQUARESIZE + SQUARESIZE / 2)), RADIUS)
            elif board[r][c] == 2:
                pygame.draw.circle(screen, YELLOW, (
                int(c * SQUARESIZE + SQUARESIZE / 2), height - int(r *
SQUARESIZE + SQUARESIZE / 2)), RADIUS)
    pygame.display.update()

board = create_board()
print_board(board)
game_over = False
turn = 0

pygame.init()

SQUARESIZE = 100

width = COLUMN_COUNT * SQUARESIZE
height = (ROW_COUNT + 1) * SQUARESIZE

size = (width, height)

RADIUS = int(SQUARESIZE / 2 - 5)

screen = pygame.display.set_mode(size)
draw_board(board)
pygame.display.update()

myfont = pygame.font.SysFont("monospace", 40)

while not game_over:

    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, SQUARESIZE))
            posx = event.pos[0]
            if turn == 0:
                pygame.draw.circle(screen, RED, (posx, int(SQUARESIZE /
2)), RADIUS)
            else:
                pygame.draw.circle(screen, YELLOW, (posx, int(SQUARESIZE
/ 2)), RADIUS)
        pygame.display.update()

        if event.type == pygame.MOUSEBUTTONDOWN:
```

```
            pygame.draw.rect(screen, BLACK, (0, 0, width, SQUARESIZE))
            if turn == 0:
                posx = event.pos[0]
                col = int(math.floor(posx / SQUARESIZE))

                if is_valid_location(board, col):
                    row = get_next_open_row(board, col)
                    drop_piece(board, row, col, 1)

                    if winning_move(board, 1):
                        label = myfont.render("Jogador vermelho
venceu!!", 1, RED)
                        screen.blit(label, (40, 10))
                        game_over = True

            else:
                posx = event.pos[0]
                col = int(math.floor(posx / SQUARESIZE))

                if is_valid_location(board, col):
                    row = get_next_open_row(board, col)
                    drop_piece(board, row, col, 2)

                    if winning_move(board, 2):
                        label = myfont.render("Jogador amarelo venceu!!",
1, YELLOW)
                        screen.blit(label, (40, 10))
                        game_over = True

            print_board(board)
            draw_board(board)

            turn += 1
            turn = turn % 2

            if game_over:
                pygame.time.wait(3000)
```