#Connect 4 Python Code

#Keith Galli copyright

|  |  |
| --- | --- |
|  | 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): |
|  | # Check horizontal locations for win |
|  | 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 |
|  |  |
|  | # Check vertical locations for win |
|  | 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 |
|  |  |
|  | # Check positively sloped diaganols |
|  | 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 |
|  |  |
|  | # Check negatively sloped diaganols |
|  | 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", 75) |
|  |  |
|  | 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)) |
|  | #print(event.pos) |
|  | # Ask for Player 1 Input |
|  | 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("Player 1 wins!!", 1, RED) |
|  | screen.blit(label, (40,10)) |
|  | game\_over = True |
|  |  |
|  |  |
|  | # # Ask for Player 2 Input |
|  | 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("Player 2 wins!!", 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) |