import pygame

import sys

WIDTH = 800

ROWS = 20

WIN = pygame.display.set\_mode((WIDTH, WIDTH))

alive\_color\_WHITE = (255, 255, 255)

dead\_color\_BLACK = (0, 0, 0)

class Game:

def \_\_init\_\_(self, row, col, width):

self.row = row

self.col = col

self.x = int(row \* width)

self.y = int(col \* width)

self.colour = WHITE

self.occupied = None

def draw(self, WIN):

pygame.draw.rect(WIN, self.colour, (self.x, self.y, WIDTH / 8, WIDTH / 8))

def make\_grid(rows, width):

grid = []

gap = WIDTH // rows

print(gap)

for i in range(rows):

grid.append([])

for j in range(rows):

game = Game(j, i, gap)

grid[i].append(node)

return grid

def draw\_grid(win, rows, width):

gap = width // ROWS

for i in range(rows):

pygame.draw.line(win, BLACK, (0, i \* gap), (width, i \* gap))

for j in range(rows):

pygame.draw.line(win, BLACK, (j \* gap, 0), (j \* gap, width))

"""

The nodes are all white so this we need to draw the grey lines that separate all the chess tiles

from each other and that is what this function does"""

def update\_display(win, grid, rows, width):

for row in grid:

for spot in row:

spot.draw(win)

draw\_grid(win, rows, width)

pygame.display.update()

def Find\_Game(pos, WIDTH):

interval = WIDTH / ROWS

y, x = pos

rows = y // interval

columns = x // interval

return int(rows), int(columns)

def neighbour(self):

col, row = self.row, self.col

# print(row, col)

neighbours = [[row - 1, col - 1], [row - 1, col], [row - 1, col + 1],

[row, col - 1], [row, col + 1],

[row + 1, col - 1], [row + 1, col], [row + 1, col + 1], ]

actual = []

for i in neighbours:

row, col = i

if 0 <= row <= (ROWS - 1) and 0 <= col <= (ROWS - 1):

actual.append(i)

# print(row, col, actual)

return actual

def update\_grid(grid):

newgrid = []

for row in grid:

for self in row:

neighbours = neighbour(self)

count = 0

for i in neighbours:

row, col = i

if grid[row][col].colour == BLACK:

count += 1

if self.colour == BLACK:

if count == 2 or count == 3:

newgrid.append(BLACK)

else:

newgrid.append(WHITE)

else:

if count == 3:

newgrid.append(BLACK)

else:

newgrid.append(WHITE)

return newgrid

def main(WIN, WIDTH):

run = None

grid = make\_grid(ROWS, WIDTH)

while True:

pygame.time.delay(50) ##stops cpu dying

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_SPACE:

run = True

if event.type == pygame.MOUSEBUTTONDOWN:

pos = pygame.mouse.get\_pos()

row, col = Find\_Node(pos, WIDTH)

if grid[col][row].colour == WHITE:

grid[col][row].colour = BLACK

elif grid[col][row].colour == BLACK:

grid[col][row].colour = WHITE

while run:

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

run = False

#pygame.time.delay(50)

newcolours = update\_grid(grid)

count=0

for i in range(0,len(grid[0])):

for j in range(0, len(grid[0])):

grid[i][j].colour=newcolours[count]

count+=1

update\_display(WIN, grid, ROWS, WIDTH)

#run= False

update\_display(WIN, grid, ROWS, WIDTH)

main(WIN, WIDTH)