import pygame,sys,time

from pygame.locals import \*

import pygame as pg

def display\_map(width,height):

NTUmap = pygame.image.load("NTUmap.png")

NTUmap = pygame.transform.scale(NTUmap,(width,height))

screen = pygame.display.set\_mode((width,height))

screen.blit(NTUmap,(0,0))

pygame.display.flip()

def mouseClick():

while True:

for event in pygame.event.get():

if event.type == QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.MOUSEBUTTONDOWN: #when the mouse buttons are pressed

x,y = pygame.mouse.get\_pos()

print("X: %d, Y: %d" %(x,y))

if x<(900/2):

print('right')

else:

print('left')

if y<700/2:

print('top')

else:

print('bottom')

pygame.display.flip()

if \_\_name\_\_ == '\_\_main\_\_':

pygame.init()

width = 900

height = 700

NTUmap = pygame.image.load("NTUmap.png")

NTUmap = pygame.transform.scale(NTUmap,(width,height))

screen = pygame.display.set\_mode((width,height))

screen.blit(NTUmap,(0,0))

font = pygame.font.Font(None, 32) #create a new Font object: Font(object,size)->Font

clock = pygame.time.Clock() #create an object to help track time

input\_box = pygame.Rect(0, 0, 140, 32) #(left,top,width,height): store rectangle coordinates

color\_inactive = pg.Color('lightskyblue3') #color when the input\_box is inactive: blurred blue

color\_active = pg.Color('dodgerblue2') #color when the input\_box is active

color = color\_inactive #first set the color inactive

active = False

text = ''

done = False

while not done:

for event in pygame.event.get():

if event.type == pygame.QUIT: #when the user clicks the window's "X" button

done = True

if event.type == pygame.MOUSEBUTTONDOWN: #when the mouse buttons are pressed

# If the user clicked on the input\_box rect.

if input\_box.collidepoint(event.pos): #test if a point is inside a rectangle: collidepoint(x,y)->bool

#toggle the active value

active = True

else:

active =False

#change the current color

color = color\_active if active else color\_inactive

if event.type == pygame.KEYDOWN:

if active:

if event.key == pg.K\_RETURN: #press Enter

print(text)

text = ''

elif event.key == pg.K\_BACKSPACE:

text = text[:-1]

else:

text += event.unicode #add the character that has already translated from keypress

# Render the current text.

txt\_surface = font.render(text, True, color)

# Resize the box if the text is too long.

width = max(200, txt\_surface.get\_width()+10)

input\_box.w = width

# Blit the text.

screen.blit(txt\_surface, (input\_box.x+5, input\_box.y+5))

# Blit the input\_box rect.

pygame.draw.rect(screen, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

pygame.quit()