# This is a sample Python script.

import arcade

import pygame

import ballsort

from ballsort import colour

white = (255, 255, 255)

black = (0, 0, 0)

red = (255, 0, 0)

green = (0, 255, 0)

blue = (0, 0, 255)

colours=[red,green,blue]

# Press the green button in the gutter to run the script.

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

#start game

pygame.init()

#display

gameDisplay = pygame.display.set\_mode((650, 500))

gameDisplay.fill(black)

#variables

x=100

y=150

i=0

#setting up containers

containters=[]

while i < 5:

containters.append(ballsort.container)

containters[i].object = pygame.draw.rect(gameDisplay, white, (x, y, 50, 200))

#print(containters[i].object.x)

x = x + 100

i= i + 1

#setting up balls

x=125

y=175

a = 1

containerNo = len(containters)

colourList = ballsort.colour(colours, containerNo)

balls = ballsort.balls(colourList)

placement = ballsort.placement(containerNo, balls)

cL=[]

for i in colourList:

ballsort.ball = pygame.draw.circle(gameDisplay, i, (x, y), 15)

cL.append(ballsort.ball)

y=y+50

if a%4 == 0 :

x = x+100

y = 175

a=a+1

#test

'''

for i in placement:

print("this container has balls with the colours: ")

for j in i.balls:

print(j.colour)

print()

'''

print(containters[1].object.x)

rects=cL

selected = None

#quit game

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

quit()

elif event.type == pygame.MOUSEBUTTONDOWN:

if event.button == 1:

for i, r in enumerate(rects):

# Pythagoras a^2 + b^2 = c^2

dx = r.centerx - event.pos[0] # a

dy = r.centery - event.pos[1] # b

distance\_square = dx \*\* 2 + dy \*\* 2 # c^2

if distance\_square <= 15 \*\* 2: # c^2 <= radius^2

selected = i

selected\_offset\_x = r.x - event.pos[0]

selected\_offset\_y = r.y - event.pos[1]

elif event.type == pygame.MOUSEBUTTONUP:

if event.button == 1:

selected = None

elif event.type == pygame.MOUSEMOTION:

if selected is not None: # selected can be `0` so `is not None` is required

# move object

rects[selected].x = event.pos[0] + selected\_offset\_x

rects[selected].y = event.pos[1] + selected\_offset\_y

gameDisplay.fill((0,0,0))

for j in range(len(containters)):

pygame.draw.rect(gameDisplay, white, (containters[j].object.x, 150, 50, 200))

for i in range(len(colourList)):

pygame.draw.circle(gameDisplay,colourList[i], rects[i].center, 10)

pygame.display.update()

# See PyCharm help at https://www.jetbrains.com/help/pycharm/