"""

Label demonstation using a class,

Express Line programming book, p. 247ff

"""

import pygame

pygame.init()

# Set the width and height of the screen [width, height]

size = (800, 600)

screen = pygame.display.set\_mode(size)

# Set up the Label class

class Label(pygame.sprite.Sprite):

"""Label class (simplest version)

Attributes:

font: any Pygame font or SysFont obj

text: the text to display on screen

center: desired position of label's center (x,y)

"""

def \_\_init\_\_(self):

pygame.sprite.Sprite.\_\_init\_\_(self)

self.font = pygame.font.SysFont("Arial", 25)

self.text = ""

self.center = (320, 240)

self.text\_color = (255, 0, 0)

def update(self):

self.image = self.font.render(self.text, 1, self.text\_color)

self.rect = self.image.get\_rect()

self.rect.center = self.center

def main():

pygame.display.set\_caption("My Basic Labels Demo")

background = pygame.Surface(screen.get\_size())

background.fill((255, 255, 255))

screen.blit(background, (0,0))

label1 = Label()

label2 = Label()

label\_event = Label()

all\_sprites = pygame.sprite.Group(label1, label2, label\_event)

label1.text = "Hi! This is Label #1 speaking!"

label1.center = (200, 100)

label2.text = "Label #2 here! Ignore Label #1!"

label2.center = (450, 400)

# Used to manage how fast the screen updates

clock = pygame.time.Clock()

# Loop until the user clicks the close button.

keep\_going = True

# -------- Main Program Loop -----------

while keep\_going:

# --- Limit to 30 frames per second

clock.tick(30)

# --- Main event loop

for event in pygame.event.get():

if event.type == pygame.QUIT:

keep\_going = False

elif event.type == pygame.MOUSEMOTION:

(mouse\_x, mouse\_y) = pygame.mouse.get\_pos()

label\_event.text = "mouse:" + str((mouse\_x, mouse\_y))

elif event.type == pygame.MOUSEBUTTONDOWN:

label\_event.text = "button press!"

elif event.type == pygame.KEYDOWN:

label\_event.text = "key down!"

all\_sprites.clear(screen, background)

all\_sprites.update()

all\_sprites.draw(screen)

pygame.display.flip()

if \_\_name\_\_ == "\_\_main\_\_":

main()