# Sample Python code for displaying GAME OVER message on screen

import pygame

# Define some colors

BLACK = (0, 0, 0)

WHITE = (255, 255, 255)

GREEN = (0, 255, 0)

pygame.init()

# Set the height and width of the screen

size = [700, 500]

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

pygame.display.set\_caption("Game Over Example")

# Loop until the user clicks the close button.

done = False

# Used to manage how fast the screen updates

clock = pygame.time.Clock()

# Starting position of the rectangle

rect\_x = 50

rect\_y = 50

# Speed and direction of rectangle

rect\_change\_x = 5

rect\_change\_y = 5

# This is a font we use to draw text on the screen (size 36)

font = pygame.font.Font(None, 36)

# Use this boolean variable to trigger if the game is over.

game\_over = False

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

while not done:

# --- Event Processing

for event in pygame.event.get():

if event.type == pygame.QUIT:

done = True

# We will use a mouse-click to signify when the game is

# over. Replace this, and set game\_over to true in your

# own game when you know the game is over. (Like lives==0)

elif event.type == pygame.MOUSEBUTTONDOWN:

game\_over = True

# --- Game Logic

# Only move and process game logic if the game isn't over.

if not game\_over:

# Move the rectangle starting point

rect\_x += rect\_change\_x

rect\_y += rect\_change\_y

# Bounce the ball if needed

if rect\_y > 450 or rect\_y < 0:

rect\_change\_y = rect\_change\_y \* -1

if rect\_x > 650 or rect\_x < 0:

rect\_change\_x = rect\_change\_x \* -1

# --- Draw the frame

# Set the screen background

screen.fill(BLACK)

# Draw the rectangle

pygame.draw.rect(screen, GREEN, [rect\_x, rect\_y, 50, 50])

if game\_over:

# If game over is true, draw game over

text = font.render("Game Over", True, WHITE)

text\_rect = text.get\_rect()

text\_x = screen.get\_width() / 2 - text\_rect.width / 2

text\_y = screen.get\_height() / 2 - text\_rect.height / 2

screen.blit(text, [text\_x, text\_y])

else:

# If game isn't over, draw this stuff.

text = font.render("Click to end game", True, WHITE)

text\_rect = text.get\_rect()

text\_x = screen.get\_width() / 2 - text\_rect.width / 2

text\_y = screen.get\_height() / 2 - text\_rect.height / 2

screen.blit(text, [text\_x, text\_y])

# Limit frames per second

clock.tick(60)

# Go ahead and update the screen with what we've drawn.

pygame.display.flip()

# Be IDLE friendly. If you forget this line, the program will 'hang'

# on exit.

pygame.quit()