"""

This is a Worlds hardest game style game, where you have to reach the green area

collect all of the yellow balls and avoid the blue ones.It includes a leaderboard,

instructions page, and 5 levels. The data on the leaderboard is saved into a file in

the same folder (Leaderboard.json) and is retrieved the next time the program is opened. So that the

leaders stay on the board.This game was created by Dominik Alkhovik.

Please write down any glitches or bugs that could be fixed in the next block comment

"""

"""

Bugs:

"""

import pygame # Importing Modules

import sys

import math

import json

import os

pygame.init() # Initialise the game engine

from pygame.locals import \*

# Define some colors

BLACK = ( 0, 0, 0)

WHITE = ( 255, 255, 255)

GREEN = ( 0, 255, 0)

RED = ( 255, 0, 0)

BLUE = ( 0, 0, 255)

L\_BLUE = ( 46, 134, 193)

YELLOW = ( 230, 230, 0)

PURPLE = ( 185, 76, 225)

LL\_BLUE = ( 51, 204, 204)

# Opening and setting the window size

Size = (800, 600)

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

pygame.display.set\_caption("The Worlds Hardest Game")

# Loop until the user clicks the close button

done = False

# Used to manage how fast the screen updates

clock = pygame.time.Clock()

x\_speed = 0

y\_speed = 0

x\_coord = 10

y\_coord = 10

Page = 0

Level = 1

Deaths = 0

L\_font = pygame.font.SysFont('Calibri', 70, True, False)#initialise fonts/sizes

font = pygame.font.SysFont('Calibri', 60, True, False)

font2 = pygame.font.SysFont('Calibri', 30, True, False)

font3 = pygame.font.SysFont('Calibri', 15, True, False)

Music = pygame.mixer.Sound("Music.ogg")

Background = pygame.image.load("background.png")

#---- Leader Board related code --------

Letter = 1

No1 = ""

No2 = ""

No3 = ""

No4 = ""

No5 = ""

No6 = ""

No7 = ""

No8 = ""

leader\_board = []

PlayerName = ""

#----- Level related code ---------

Lv1Yellow = False

Lv2Yellow = [0,0,0]

Lv4Yellow = False

Lv5Yellow = [1,0,0,0,0,0,0]

Lv1 = []

Lv2 = []

Lv3 = []

Lv4 = []

Lv5 = []

Lv5c = []

Count = 0

class BlueBall:

def \_\_init\_\_(self, x, y, direction,xl,xr,yt,yb,thick,speed):

self.x = x

self.y = y

self.direction = direction

self.xl = xl

self.xr = xr

self.yt = yt

self.yb = yb

self.thick = thick

self.speed = speed

def move(self):

if self.y <= self.yt:

self.direction = 2

elif self.y >= self.yb:

self.direction = 0

elif self.x <= self.xl:

self.direction = 1

elif self.x >= self.xr:

self.direction = 3

if self.direction == 0:

self.y -= self.speed

elif self.direction == 1:

self.x += self.speed

elif self.direction == 2:

self.y += self.speed

elif self.direction == 3:

self.x -= self.speed

pygame.draw.ellipse(screen, BLUE, [self.x,self.y,self.thick,self.thick])

class BlueBallCircle:

def \_\_init\_\_(self, x, y, direction,amp,step,size):

self.x = x

self.y = y

self.direction = direction

self.amp = amp

self.step = step

self.size = size

def drawcircle(self):

if self.direction == 1:

self.step -= 0.017

self.step %= 2 \* math.pi

if self.direction == 2:

self.step += 0.017

self.step %= 2 \* math.pi

self.xPos = int(math.cos(self.step) \* self.amp)

self.yPos = int(-1 \* math.sin(self.step) \* self.amp)

pygame.draw.ellipse(screen, BLUE, [int(self.xPos) + self.x, int(self.yPos) + self.y, self.size, self.size])

def square(x):

return x\*x

def find\_letter(letter):

global No1

global No2

global No3

global No4

global No5

global No6

global No7

global No8

if Letter == 1:

No1 = letter

if Letter == 2:

No2 = letter

if Letter == 3:

No3 = letter

if Letter == 4:

No4 = letter

if Letter == 5:

No5 = letter

if Letter == 6:

No6 = letter

if Letter == 7:

No7 = letter

if Letter == 8:

No8 = letter

#--------------Main event loop -----------

while not done:

Music.play()

for event in pygame.event.get():

if event.type == pygame.QUIT:

done = True

# User pressed down on a key

elif event.type == pygame.KEYDOWN:

if event.key == pygame.K\_LEFT or event.key == pygame.K\_KP4:

x\_speed = -1

elif event.key == pygame.K\_RIGHT or event.key == pygame.K\_KP6:

x\_speed = 1

elif event.key == pygame.K\_UP or event.key == pygame.K\_KP8:

y\_speed = -1

elif event.key == pygame.K\_DOWN or event.key == pygame.K\_KP2:

y\_speed = 1

# User let up on a key

elif event.type == pygame.KEYUP:

if event.key == pygame.K\_LEFT or event.key == pygame.K\_RIGHT:

x\_speed = 0

elif event.key == pygame.K\_UP or event.key == pygame.K\_DOWN:

y\_speed = 0

elif event.key == pygame.K\_c:

if Level < 5:

Level = Level+1

else:

Page = 5

# --- Game logic Should go here

x\_coord += x\_speed

y\_coord += y\_speed

pos = pygame.mouse.get\_pos()

x = pos[0]

y = pos[1]

# First , clear the screen to White

screen.fill(WHITE)

#------------ PAGE 0 ------------------------

if Page == 0:

screen.blit(Background, (0, 0))

Title = L\_font.render("The Worlds", True, BLUE)

Title2 = L\_font.render("Hardest Game", True, BLUE)

screen.blit(Title, [165, 60])

screen.blit(Title2, [270, 130])

Play = font.render("Play ", True, RED)

Instr = font.render("Instructions ", True, L\_BLUE)

Leader = font.render("Leader Board ", True, GREEN)

screen.blit(Play, [330, 250])

screen.blit(Instr, [237, 330])

screen.blit(Leader, [219, 410])

By = font2.render("By Dominik Alkhovik", True, BLACK)

Version = font3.render("Version 1.0.1", True, BLACK)

screen.blit(By, [20, 550])

screen.blit(Version, [700, 580])

pygame.mouse.set\_visible(True)

if event.type == pygame.MOUSEBUTTONUP and x > 235 and x < 535 and y > 250 and y < 300:

Level = 1

Deaths = 0

Letter = 1

No1 = ""

No2 = ""

No3 = ""

No4 = ""

No5 = ""

No6 = ""

No7 = ""

No8 = ""

Page = 1

x\_coord = 40

y\_coord = 280

if event.type == pygame.MOUSEBUTTONUP and x > 225 and x < 545 and y > 330 and y < 380:

Page = 2

if event.type == pygame.MOUSEBUTTONUP and x > 215 and x < 555 and y > 410 and y < 460:

Page = 3

#------------------------ PAGE 1 --------------------------------

if Page == 1:

if Level == 1:

if Lv1 == []:

for i in range(12):

direction = 0

if i%2 == 1:

direction = 2

Lv1.append(BlueBall(105+(i\*50),280,direction,0,800,50,510,40,2))

for i in Lv1:

i.move()

pygame.draw.rect(screen, LL\_BLUE, [0,0,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [0,390,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,0,800,50])

pygame.draw.rect(screen, LL\_BLUE, [700,0,100,210])

pygame.draw.rect(screen, LL\_BLUE, [780,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [700,390,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,550,800,50])

pygame.draw.rect(screen, GREEN, [ 700,210,80,180])

pygame.draw.rect(screen, BLACK, [760,10,10,30])

pygame.draw.rect(screen, BLACK, [777,10,10,30])

DeathsS = font2.render("Deaths:", True, BLACK)

DeathsInt = font2.render(str(Deaths), True, BLACK)

screen.blit(DeathsS, [600, 13])

screen.blit(DeathsInt, [700, 13])

pygame.draw.rect(screen, RED, [x\_coord,y\_coord,40,40])

pygame.draw.line(screen, BLACK, [100,50],[700, 50],5)

pygame.draw.line(screen, BLACK, [100,550],[700, 550],5)

pygame.draw.line(screen, BLACK, [100,50],[100, 210],5)

pygame.draw.line(screen, BLACK, [100,550],[100, 390],5)

pygame.draw.line(screen, BLACK, [100,210],[20, 210],5)

pygame.draw.line(screen, BLACK, [20,210],[20, 390],5)

pygame.draw.line(screen, BLACK, [20,390],[100, 390],5)

pygame.draw.line(screen, BLACK, [100,50],[700, 50],5)

pygame.draw.line(screen, BLACK, [700,50],[700, 210],5)

pygame.draw.line(screen, BLACK, [700,210],[780, 210],5)

pygame.draw.line(screen, BLACK, [780,210],[780, 390],5)

pygame.draw.line(screen, BLACK, [780,390],[700, 390],5)

pygame.draw.line(screen, BLACK, [700,390],[700, 550],5)

if event.type == pygame.MOUSEBUTTONUP and x > 750 and x < 800 and y > 0 and y < 50:#Pause button clicked

Page = 4

if x\_coord < 20:

x\_coord = 20

if x\_coord > 740:

x\_coord = 740

if y\_coord < 50:

y\_coord = 50

if y\_coord > 510:

y\_coord = 510

if y\_coord < 210 and y\_coord > 208:

if x\_coord < 100:

y\_coord = 210

elif y\_coord > 350 and y\_coord < 352:

if x\_coord < 100:

y\_coord = 350

elif x\_coord <= 100:

if y\_coord < 210 or y\_coord > 350:

x\_coord = 100

if y\_coord < 210 and y\_coord > 208:

if x\_coord > 660:

y\_coord = 210

elif y\_coord > 350 and y\_coord < 352:

if x\_coord > 660:

y\_coord = 350

elif x\_coord >= 660:

if y\_coord < 210 or y\_coord > 350:

x\_coord = 660

collided = False

for ball in Lv1:

if square(x\_coord - ball.x) + square(y\_coord - ball.y) <= 1600:

collided = True

break

if collided == True:

Deaths += 1

x\_coord =10

y\_coord =280

Lv1Yellow = False

if square(x\_coord - 375) + square(y\_coord - 280) <= 1600:

Lv1Yellow = True

if Lv1Yellow == False:

pygame.draw.ellipse(screen, YELLOW, [375,280,40,40])

if x\_coord > 660 and y\_coord > 210 and y\_coord < 350 and Lv1Yellow == True:

Lv1Yellow = False

x\_coord = 350

y\_coord = 50

Level = 2

Page = 6

#--------------------------- LEVEL 2 --------------------------------------

if Level == 2:

if Lv2 == []:

Lv2.append(BlueBallCircle(380,330,1,50,0.005,40))

Lv2.append(BlueBallCircle(380,330,1,100,0.005,40))

Lv2.append(BlueBallCircle(380,330,1,150,0.005,40))

Lv2.append(BlueBallCircle(380,330,1,-50,0.005,40))

Lv2.append(BlueBallCircle(380,330,1,-100,0.005,40))

Lv2.append(BlueBallCircle(380,330,1,-150,0.005,40))

for i in Lv2:

i.drawcircle()

pygame.draw.ellipse(screen, BLUE, [380, 330, 40, 40])

pygame.draw.rect(screen, LL\_BLUE, [0,0,800,50])

pygame.draw.rect(screen, LL\_BLUE, [0,550,800,50])

pygame.draw.rect(screen, LL\_BLUE, [0,0,200,600])

pygame.draw.rect(screen, LL\_BLUE, [600,0,200,600])

pygame.draw.rect(screen, LL\_BLUE, [450,0,200,150])

pygame.draw.rect(screen, LL\_BLUE, [500,0,100,200])

pygame.draw.rect(screen, LL\_BLUE, [550,0,50,250])

pygame.draw.rect(screen, LL\_BLUE, [550,450,200,50])

pygame.draw.rect(screen, LL\_BLUE, [500,500,800,50])

pygame.draw.rect(screen, LL\_BLUE, [200,0,150,150])

pygame.draw.rect(screen, LL\_BLUE, [200,0,100,200])

pygame.draw.rect(screen, LL\_BLUE, [200,0,50,250])

pygame.draw.rect(screen, LL\_BLUE, [200,450,50,50])

pygame.draw.rect(screen, LL\_BLUE, [200,500,100,50])

pygame.draw.rect(screen, GREEN, [ 100,300,100,100])

pygame.draw.rect(screen, BLACK, [760,10,10,30])

pygame.draw.rect(screen, BLACK, [777,10,10,30])

DeathsS = font2.render("Deaths:", True, BLACK)

DeathsInt = font2.render(str(Deaths), True, BLACK)

screen.blit(DeathsS, [600, 13])

screen.blit(DeathsInt, [700, 13])

pygame.draw.rect(screen, RED, [x\_coord,y\_coord,40,40])

pygame.draw.line(screen, BLACK, [300,550],[500,550],5)

pygame.draw.line(screen, BLACK, [300,552],[300,500],5)

pygame.draw.line(screen, BLACK, [250,500],[300,500],5)

pygame.draw.line(screen, BLACK, [250,500],[250,450],5)

pygame.draw.line(screen, BLACK, [200,450],[250,450],5)

pygame.draw.line(screen, BLACK, [200,450],[200,400],5)

pygame.draw.line(screen, BLACK, [100,400],[200,400],5)

pygame.draw.line(screen, BLACK, [100,400],[100,300],5)

pygame.draw.line(screen, BLACK, [200,300],[100,300],5)

pygame.draw.line(screen, BLACK, [200,300],[200,250],5)

pygame.draw.line(screen, BLACK, [250,250],[200,250],5)

pygame.draw.line(screen, BLACK, [250,250],[250,200],5)

pygame.draw.line(screen, BLACK, [300,200],[250,200],5)

pygame.draw.line(screen, BLACK, [300,200],[300,150],5)

pygame.draw.line(screen, BLACK, [350,150],[300,150],5)

pygame.draw.line(screen, BLACK, [350,150],[350,50],5)

pygame.draw.line(screen, BLACK, [450,50],[350,50],5)

pygame.draw.line(screen, BLACK, [450,50],[450,150],5)

pygame.draw.line(screen, BLACK, [500,150],[450,150],5)

pygame.draw.line(screen, BLACK, [500,150],[500,200],5)

pygame.draw.line(screen, BLACK, [550,200],[500,200],5)

pygame.draw.line(screen, BLACK, [550,200],[550,250],5)

pygame.draw.line(screen, BLACK, [600,250],[550,250],5)

pygame.draw.line(screen, BLACK, [600,250],[600,450],5)

pygame.draw.line(screen, BLACK, [550,450],[600,450],5)

pygame.draw.line(screen, BLACK, [550,450],[550,500],5)

pygame.draw.line(screen, BLACK, [500,500],[550,500],5)

pygame.draw.line(screen, BLACK, [500,500],[500,550],5)

if x\_coord <= 100:

x\_coord = 100

if x\_coord >= 560:

x\_coord = 560

if y\_coord <= 50:

y\_coord = 50

if y\_coord >= 510:

y\_coord = 510

for i in range(4):

if y\_coord < 300-(i\*50) and y\_coord > 298-(i\*50):

if x\_coord < 200+(i\*50):

y\_coord = 300-(i\*50)

elif y\_coord > 360+(i\*50) and y\_coord < 362+(i\*50):

if x\_coord < 200+(i\*50):

y\_coord = 360+(i\*50)

elif x\_coord <= 200+(i\*50):

if y\_coord < 300-(i\*50) or y\_coord > 360+(i\*50):

x\_coord = 200+(i\*50)

if y\_coord < 150 and y\_coord > 148:

if x\_coord > 410:

y\_coord = 150

elif x\_coord >= 410:

if y\_coord < 150:

x\_coord = 410

for i in range(2):

if y\_coord < 200+(i\*50) and y\_coord > 198+(i\*50):

if x\_coord > 460+(i\*50):

y\_coord = 200+(i\*50)

elif y\_coord > 460-(i\*50) and y\_coord < 462-(i\*50):

if x\_coord > 460+(i\*50):

y\_coord = 460-(i\*50)

elif x\_coord >= 460+(i\*50):

if y\_coord < 200+(i\*50) or y\_coord > 460-(i\*50):

x\_coord = 460+(i\*50)

if event.type == pygame.MOUSEBUTTONUP and x > 750 and x < 800 and y > 0 and y < 50:#Pause button clicked

Page = 4

if Lv2Yellow[0] == 0:

pygame.draw.ellipse(screen, YELLOW, [380, 200,40,40])

if Lv2Yellow[1] == 0:

pygame.draw.ellipse(screen, YELLOW, [500, 330,40,40])

if Lv2Yellow[2] == 0:

pygame.draw.ellipse(screen, YELLOW, [380, 450,40,40])

if square(x\_coord - 380) + square(y\_coord - 200) <= 1600:

Lv2Yellow[0] = 1

if square(x\_coord - 500) + square(y\_coord - 330) <= 1600:

Lv2Yellow[1] = 1

if square(x\_coord - 380) + square(y\_coord - 450) <= 1600:

Lv2Yellow[2] = 1

collided = False

for ball in Lv2:

if square(x\_coord - (ball.xPos + ball.x)) + square(y\_coord - (ball.yPos + ball.y)) <= 1600:

collided = True

break

if collided == True:

Deaths += 1

x\_coord =380

y\_coord =40

Lv2Yellow = [0,0,0]

if x\_coord < 199 and Lv2Yellow == [1,1,1]:

x\_coord = 375

y\_coord = 95

Lv2Yellow = [0,0,0]

Level = 3

Page = 6

#----------------------------- LEVEL 3 -----------------------------------

if Level == 3:

if Lv3 == []:

for i in range(4):

direction = 1

if i%2 == 1:

direction = 3

Lv3.append(BlueBall(310,190+(i\*50),direction,275,345,0,600,25,0.5))

for i in range(3):

direction = 3

if i%2 == 1:

direction = 1

Lv3.append(BlueBall(210,340+(i\*50),direction,175,245,0,600,25,0.5))

for i in range(5):

direction = 0

if i%2 == 1:

direction = 2

Lv3.append(BlueBall(285+(i\*50),460,direction,0,800,425,495,25,0.5))

for i in range(3):

direction = 1

if i%2 == 1:

direction = 3

Lv3.append(BlueBall(560,340+(i\*50),direction,525,595,0,600,25,0.5))

for i in range(4):

direction = 3

if i%2 == 1:

direction = 1

Lv3.append(BlueBall(460,190+(i\*50),direction,425,495,0,600,25,0.5))

for i in Lv3:

i.move()

pygame.draw.rect(screen, LL\_BLUE, [0,0,175,600])

pygame.draw.rect(screen, LL\_BLUE, [0,0,275,325])

pygame.draw.rect(screen, LL\_BLUE, [0,0,800,75])

pygame.draw.rect(screen, LL\_BLUE, [0,475,275,600])

pygame.draw.rect(screen, LL\_BLUE, [625,0,175,600])

pygame.draw.rect(screen, LL\_BLUE, [525,475,175,600])

pygame.draw.rect(screen, LL\_BLUE, [0,525,700,600])

pygame.draw.rect(screen, LL\_BLUE, [525,150,100,175])

pygame.draw.rect(screen, LL\_BLUE, [425,75,50,100])

pygame.draw.rect(screen, LL\_BLUE, [325,150,150,25])

pygame.draw.rect(screen, LL\_BLUE, [375,175,50,200])

pygame.draw.rect(screen, LL\_BLUE, [275,375,250,50])

pygame.draw.rect(screen, GREEN, [ 475,75,150,75])

pygame.draw.rect(screen, BLACK, [760,10,10,30])

pygame.draw.rect(screen, BLACK, [777,10,10,30])

DeathsS = font2.render("Deaths:", True, BLACK)

DeathsInt = font2.render(str(Deaths), True, BLACK)

screen.blit(DeathsS, [600, 13])

screen.blit(DeathsInt, [700, 13])

pygame.draw.rect(screen, RED, [x\_coord,y\_coord,30,30])

pygame.draw.line(screen, BLACK, [300-25,50+25],[450-25, 50+25] ,5)

pygame.draw.line(screen, BLACK, [300-25,50+25],[300-25, 300+25] ,5)

pygame.draw.line(screen, BLACK, [300-25,300+25],[200-25, 300+25],5)

pygame.draw.line(screen, BLACK, [200-25,300+25],[200-25, 450+25],5)

pygame.draw.line(screen, BLACK, [200-25,450+25],[300-25, 450+25],5)

pygame.draw.line(screen, BLACK, [300-25,450+25],[300-25, 500+25],5)

pygame.draw.line(screen, BLACK, [300-25,500+25],[550-25, 500+25],5)

pygame.draw.line(screen, BLACK, [550-25,500+25],[550-25, 450+25],5)

pygame.draw.line(screen, BLACK, [550-25,450+25],[650-25, 450+25],5)

pygame.draw.line(screen, BLACK, [650-25,450+25],[650-25, 300+25],5)

pygame.draw.line(screen, BLACK, [650-25,300+25],[550-25, 300+25],5)

pygame.draw.line(screen, BLACK, [550-25,300+25],[550-25, 125+25],5)

pygame.draw.line(screen, BLACK, [550-25,125+25],[650-25, 125+25],5)

pygame.draw.line(screen, BLACK, [650-25,125+25],[650-25, 50+25] ,5)

pygame.draw.line(screen, BLACK, [650-25,50+25],[500-25, 50+25] ,5)

pygame.draw.line(screen, BLACK, [500-25,50+25],[500-25, 150+25] ,5)

pygame.draw.line(screen, BLACK, [500-25,150+25],[450-25, 150+25],5)

pygame.draw.line(screen, BLACK, [450-25,150+25],[450-25, 350+25],5)

pygame.draw.line(screen, BLACK, [450-25,350+25],[550-25, 350+25],5)

pygame.draw.line(screen, BLACK, [550-25,350+25],[550-25, 400+25],5)

pygame.draw.line(screen, BLACK, [550-25,400+25],[300-25, 400+25],5)

pygame.draw.line(screen, BLACK, [300-25,400+25],[300-25, 350+25],5)

pygame.draw.line(screen, BLACK, [300-25,350+25],[400-25, 350+25],5)

pygame.draw.line(screen, BLACK, [400-25,350+25],[400-25, 150+25],5)

pygame.draw.line(screen, BLACK, [400-25,150+25],[350-25, 150+25],5)

pygame.draw.line(screen, BLACK, [350-25,150+25],[350-25, 125+25],5)

pygame.draw.line(screen, BLACK, [350-25,125+25],[450-25, 125+25],5)

pygame.draw.line(screen, BLACK, [450-25,125+25],[450-25, 50+25] ,5)

if x\_coord <= 175:

x\_coord = 175

if x\_coord >= 595:

x\_coord = 595

if y\_coord <= 75:

y\_coord = 75

if y\_coord >= 525-30:

y\_coord = 525-30

if x\_coord < 275 and x\_coord > 273:

if y\_coord < 325 or y\_coord > 445:

x\_coord = 275

if y\_coord < 325 and y\_coord > 323:

if x\_coord < 275 or x\_coord > 495:

y\_coord = 325

if y\_coord > 445 and y\_coord < 447:

if x\_coord < 275 or x\_coord > 495:

y\_coord = 445

if x\_coord > 495 and x\_coord < 497:

if (y\_coord < 325 and y\_coord > 120) or y\_coord > 445:

x\_coord = 495

if y\_coord > 120 and y\_coord < 122:

if x\_coord > 495 or (x\_coord > 295 and x\_coord < 430):

y\_coord = 120

if x\_coord < 475 and x\_coord > 473:

if y\_coord < 175:

x\_coord = 475

if y\_coord < 175 and y\_coord > 173:

if (x\_coord < 475 and x\_coord > 415)or (x\_coord > 295 and x\_coord < 350):

y\_coord = 175

if x\_coord < 425 and x\_coord > 423:

if y\_coord < 420:

x\_coord = 425

if x\_coord > 345 and x\_coord < 347:

if y\_coord > 170 and y\_coord < 420:

x\_coord = 345

if y\_coord > 345 and y\_coord < 347:

if x\_coord < 525 and x\_coord > 245:

y\_coord = 345

if x\_coord < 525 and x\_coord > 523:

if y\_coord > 345 and y\_coord < 425:

x\_coord = 525

if y\_coord < 425 and y\_coord > 423:

if x\_coord > 245 and x\_coord < 525:

y\_coord = 425

if x\_coord > 245 and x\_coord < 247:

if y\_coord > 345 and y\_coord < 425:

x\_coord = 245

if x\_coord > 295 and x\_coord < 297:

if y\_coord > 120 and y\_coord < 175:

x\_coord = 295

if x\_coord > 395 and x\_coord < 397:

if y\_coord < 125:

x\_coord = 395

if event.type == pygame.MOUSEBUTTONUP and x > 750 and x < 800 and y > 0 and y < 50:#Pause button clicked

Page = 4

collided = False

for ball in Lv3:

if square(x\_coord - ball.x) + square(y\_coord - ball.y) <= 850:

collided = True

break

if collided == True:

Deaths += 1

x\_coord = 375

y\_coord = 95

if x\_coord > 425 and y\_coord < 150 and y\_coord < 350:

x\_coord = 40

y\_coord = 280

Level = 4

Page = 6

#------------------------------ LEVEL 4 -------------------------------------

if Level == 4:

if Lv4 == []:

for i in range(12):

direction = 0

if i%2 == 1:

direction = 2

Lv4.append(BlueBall(105+(i\*50),280,direction,0,800,50,510,40,0.75))

for i in range(10):

direction = 1

if i%2 == 1:

direction = 3

Lv4.append(BlueBall(380,55+(i\*50),direction,100,660,0,600,40,0.75))

for i in Lv4:

i.move()

pygame.draw.rect(screen, LL\_BLUE, [0,0,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [0,390,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,0,800,50])

pygame.draw.rect(screen, LL\_BLUE, [700,0,100,210])

pygame.draw.rect(screen, LL\_BLUE, [780,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [700,390,100,210])

pygame.draw.rect(screen, LL\_BLUE, [0,550,800,50])

pygame.draw.rect(screen, GREEN, [ 700,210,80,180])

pygame.draw.rect(screen, BLACK, [760,10,10,30])

pygame.draw.rect(screen, BLACK, [777,10,10,30])

DeathsS = font2.render("Deaths:", True, BLACK)

DeathsInt = font2.render(str(Deaths), True, BLACK)

screen.blit(DeathsS, [600, 13])

screen.blit(DeathsInt, [700, 13])

pygame.draw.rect(screen, RED, [x\_coord,y\_coord,40,40])

pygame.draw.line(screen, BLACK, [100,50],[700, 50],5)

pygame.draw.line(screen, BLACK, [100,550],[700, 550],5)

pygame.draw.line(screen, BLACK, [100,50],[100, 210],5)

pygame.draw.line(screen, BLACK, [100,550],[100, 390],5)

pygame.draw.line(screen, BLACK, [100,210],[20, 210],5)

pygame.draw.line(screen, BLACK, [20,210],[20, 390],5)

pygame.draw.line(screen, BLACK, [20,390],[100, 390],5)

pygame.draw.line(screen, BLACK, [100,50],[700, 50],5)

pygame.draw.line(screen, BLACK, [700,50],[700, 210],5)

pygame.draw.line(screen, BLACK, [700,210],[780, 210],5)

pygame.draw.line(screen, BLACK, [780,210],[780, 390],5)

pygame.draw.line(screen, BLACK, [780,390],[700, 390],5)

pygame.draw.line(screen, BLACK, [700,390],[700, 550],5)

if event.type == pygame.MOUSEBUTTONUP and x > 750 and x < 800 and y > 0 and y < 50:#Pause button clicked

Page = 4

if x\_coord < 20:

x\_coord = 20

if x\_coord > 740:

x\_coord = 740

if y\_coord < 50:

y\_coord = 50

if y\_coord > 510:

y\_coord = 510

if y\_coord < 210 and y\_coord > 208:

if x\_coord < 100:

y\_coord = 210

elif y\_coord > 350 and y\_coord < 352:

if x\_coord < 100:

y\_coord = 350

elif x\_coord <= 100:

if y\_coord < 210 or y\_coord > 350:

x\_coord = 100

if y\_coord < 210 and y\_coord > 208:

if x\_coord > 660:

y\_coord = 210

elif y\_coord > 350 and y\_coord < 352:

if x\_coord > 660:

y\_coord = 350

elif x\_coord >= 660:

if y\_coord < 210 or y\_coord > 350:

x\_coord = 660

collided = False

for ball in Lv4:

if square(x\_coord - ball.x) + square(y\_coord - ball.y) <= 1600:

collided = True

break

if collided == True:

Deaths += 1

x\_coord = 40

y\_coord = 280

Lv4Yellow = False

if square(x\_coord - 375) + square(y\_coord - 280) <= 1600:

Lv4Yellow = True

if Lv4Yellow == False:

pygame.draw.ellipse(screen, YELLOW, [375,280,40,40])

if x\_coord > 660 and y\_coord > 210 and y\_coord < 350 and Lv4Yellow == True:

Lv4Yellow = False

x\_coord = 40

y\_coord = 115

Level = 5

Page = 6

#------------------------------ LEVEL 5 ----------------------------------------

if Level == 5:

if Lv5 == []:

for i in range(8):

Lv5.append(BlueBall(150+(i\*30),58+(i\*10),0,0

,800,50,175,25,1))

for i in range(5):

direction = 1

if i%2 == 1:

direction = 3

Lv5.append(BlueBall(620,50+(i\*30),direction,475,755,0,600,25,0.5))

for i in range(5):

direction = 0

if i%2 == 1:

direction = 2

Lv5.append(BlueBall(640+(i\*30),270,direction,0,800,50,465,25,0.5))

for i in range(6):

Lv5.append(BlueBall(125+(i\*30),457-(i\*10),0,0,800,330,465,25,1))

if Lv5c == []:

for i in range(4):

direction = 1

if i%2 == 1:

direction = 2

Lv5c.append(BlueBallCircle(468,398,direction,128-(i\*32),0.2,25))

for i in range(5):

direction = 1

if i%2 == 1:

direction = 2

Lv5c.append(BlueBallCircle(468,398,direction,0-(i\*32),0.2,25))

for i in Lv5c:

i.drawcircle()

for i in Lv5:

i.move()

pygame.draw.rect(screen, LL\_BLUE, [0,0,800,50])

pygame.draw.rect(screen, LL\_BLUE, [0,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [780,0,20,600])

pygame.draw.rect(screen, LL\_BLUE, [0,570,800,40])

pygame.draw.rect(screen, LL\_BLUE, [0,200,640,50])

pygame.draw.rect(screen, LL\_BLUE, [600,200,40,130])

pygame.draw.rect(screen, LL\_BLUE, [560,200,40,90])

pygame.draw.rect(screen, LL\_BLUE, [0,250,360,80])

pygame.draw.rect(screen, LL\_BLUE, [0,250,400,40])

pygame.draw.rect(screen, LL\_BLUE, [0,490,360,40])

pygame.draw.rect(screen, LL\_BLUE, [0,530,400,40])

pygame.draw.rect(screen, LL\_BLUE, [600,490,300,80])

pygame.draw.rect(screen, LL\_BLUE, [560,530,40,40])

pygame.draw.rect(screen, GREEN, [ 20,330,60,160])

pygame.draw.rect(screen, BLACK, [760,10,10,30])

pygame.draw.rect(screen, BLACK, [777,10,10,30])

DeathsS = font2.render("Deaths:", True, BLACK)

DeathsInt = font2.render(str(Deaths), True, BLACK)

screen.blit(DeathsS, [600, 13])

screen.blit(DeathsInt, [700, 13])

pygame.draw.rect(screen, RED, [x\_coord,y\_coord,25,25])

pygame.draw.line(screen, BLACK, [20,50],[780,50],5)

pygame.draw.line(screen, BLACK, [780,490],[780,50],5)

pygame.draw.line(screen, BLACK, [20,200],[640,200],5)

pygame.draw.line(screen, BLACK, [20,50],[20,200],5)

pygame.draw.line(screen, BLACK, [640,200],[640,330],5)

pygame.draw.line(screen, BLACK, [640,330],[600,330],5)

pygame.draw.line(screen, BLACK, [780,490],[600,490],5)

pygame.draw.line(screen, BLACK, [600,530],[600,490],5)

pygame.draw.line(screen, BLACK, [600,530],[560,530],5)

pygame.draw.line(screen, BLACK, [560,570],[560,530],5)

pygame.draw.line(screen, BLACK, [560,570],[400,570],5)

pygame.draw.line(screen, BLACK, [400,530],[400,570],5)

pygame.draw.line(screen, BLACK, [400,530],[360,530],5)

pygame.draw.line(screen, BLACK, [360,490],[360,530],5)

pygame.draw.line(screen, BLACK, [360,490],[20,490],5)

pygame.draw.line(screen, BLACK, [20,330],[20,490],5)

pygame.draw.line(screen, BLACK, [20,330],[360,330],5)

pygame.draw.line(screen, BLACK, [360,290],[360,330],5)

pygame.draw.line(screen, BLACK, [360,290],[400,290],5)

pygame.draw.line(screen, BLACK, [400,250],[400,290],5)

pygame.draw.line(screen, BLACK, [400,250],[560,250],5)

pygame.draw.line(screen, BLACK, [560,290],[560,250],5)

pygame.draw.line(screen, BLACK, [560,290],[600,290],5)

pygame.draw.line(screen, BLACK, [600,330],[600,290],5)

if event.type == pygame.MOUSEBUTTONUP and x > 750 and x < 800 and y > 0 and y < 50:#Pause button clicked

Page = 4

if x\_coord <= 20:

x\_coord = 20

if x\_coord >= 755:

x\_coord = 755

if y\_coord <= 50:

y\_coord = 50

if y\_coord >= 545:

y\_coord = 545

if y\_coord > 150+25 and y\_coord < 152+25:

if x\_coord < 640:

y\_coord = 150+25

if x\_coord < 640 and x\_coord > 638:

if y\_coord > 175 and y\_coord < 330:

x\_coord = 640

if y\_coord < 330 and y\_coord > 328:

if x\_coord < 360 or (x\_coord > 575 and x\_coord < 640):

y\_coord = 330

if y\_coord > 465 and y\_coord < 467:

if x\_coord < 360 or x\_coord > 575:

y\_coord = 465

if y\_coord < 250 and y\_coord > 248:

if x\_coord < 640:

y\_coord = 250

if x\_coord < 360 and x\_coord > 358:

if (y\_coord < 330 and y\_coord > 200) or y\_coord > 465:

x\_coord = 360

if x\_coord < 400 and x\_coord > 398:

if (y\_coord < 290 and y\_coord > 200) or y\_coord > 505:

x\_coord = 400

if x\_coord > 535 and x\_coord < 537:

if (y\_coord < 290 and y\_coord > 200) or y\_coord > 505:

x\_coord = 535

if x\_coord > 575 and x\_coord < 577:

if (y\_coord < 330 and y\_coord > 200) or y\_coord > 465:

x\_coord = 575

if y\_coord < 290 and y\_coord > 288:

if x\_coord < 400 or (x\_coord > 535 and x\_coord < 640):

y\_coord = 290

if y\_coord > 505 and y\_coord < 507:

if x\_coord < 400 or x\_coord > 535:

y\_coord = 505

if Lv5Yellow[1] == 0:

pygame.draw.ellipse(screen, YELLOW, [750,55,25,25])

if Lv5Yellow[2] == 0:

pygame.draw.ellipse(screen, YELLOW, [750,460,25,25])

if Lv5Yellow[3] == 0:

pygame.draw.ellipse(screen, YELLOW, [365,295,25,25])

if Lv5Yellow[4] == 0:

pygame.draw.ellipse(screen, YELLOW, [570,295,25,25])

if Lv5Yellow[5] == 0:

pygame.draw.ellipse(screen, YELLOW, [570,500,25,25])

if Lv5Yellow[6] == 0:

pygame.draw.ellipse(screen, YELLOW, [365,500,25,25])

if square(x\_coord - 750) + square(y\_coord - 55) <= 625:

Lv5Yellow[1] = 1

if square(x\_coord - 750) + square(y\_coord - 460) <= 625:

Lv5Yellow[2] = 1

if square(x\_coord - 365) + square(y\_coord - 295) <= 625:

Lv5Yellow[3] = 1

if square(x\_coord - 570) + square(y\_coord - 295) <= 625:

Lv5Yellow[4] = 1

if square(x\_coord - 570) + square(y\_coord - 500) <= 625:

Lv5Yellow[5] = 1

if square(x\_coord - 365) + square(y\_coord - 500) <= 625:

Lv5Yellow[6] = 1

collided = False

for ball in Lv5:

if square(x\_coord - ball.x) + square(y\_coord - ball.y) <= 600:

collided = True

break

for ball in Lv5c:

if square(x\_coord - (ball.xPos + ball.x)) + square(y\_coord - (ball.yPos + ball.y)) <= 600:

collided = True

break

if collided == True:

Deaths += 1

x\_coord = 40

y\_coord = 115

Lv5Yellow = [1,0,0,0,0,0,0]

if x\_coord < 80 and y\_coord > 320 and Lv5Yellow == [1,1,1,1,1,1,1]:

Page = 5

#----------------------------------- PAGE 2 ---------------------------------------

if Page == 2:

Instructions = L\_font.render("Instructions", True, L\_BLUE)

Ins = font2.render("You are the red square. Avoid the blue circles and collect the ", True, BLACK)

Ins2 = font2.render("yellow circles. Once you have collected all of the yellow",True, BLACK)

Ins3 = font2.render("circles, move to the green beacon to complete the level.",True, BLACK)

Ins4 = font2.render("You must complete all of the levels to submit your score ",True, BLACK)

Ins5 = font2.render("which is how many times you died. The fewer the better. ",True, BLACK)

Ins6 = font.render("Good Luck",True, BLACK)

Back = font2.render("Back", True, PURPLE)

screen.blit(Instructions, [230, 30])

screen.blit(Ins, [30, 150])

screen.blit(Ins2, [30, 200])

screen.blit(Ins3, [30, 250])

screen.blit(Ins4, [30, 300])

screen.blit(Ins5, [30, 350])

screen.blit(Ins6, [30, 450])

screen.blit(Back, [720, 550])

if event.type == pygame.MOUSEBUTTONUP and x > 710 and x < 800 and y > 540 and y < 600:

Page = 0

#----------------------------------- PAGE 3 -------------------------------------

if Page == 3:

if os.path.isfile('LeaderBoard.json'):

file = open('LeaderBoard.json', 'r')

leader\_board = json.loads(file.read())

file.close()

else:

file = open('LeaderBoard.json', 'w+')

file.write("[]")

file.close()

Leaderboard = L\_font.render("Leaderboard",True, GREEN)

Pos = font2.render("Pos", True, BLACK)

one = font2.render("1st", True, BLACK)

two = font2.render("2nd", True, BLACK)

three = font2.render("3rd", True, BLACK)

four = font2.render("4th", True, BLACK)

five = font2.render("5th", True, BLACK)

PlayerName\_w = font2.render("Player Name", True, BLACK)

Deaths\_w = font2.render("Deaths", True, BLACK)

screen.blit(Leaderboard, [210, 30])

screen.blit(Pos, [50, 150])

screen.blit(one, [50, 200])

screen.blit(two, [50, 250])

screen.blit(three, [50, 300])

screen.blit(four, [50, 350])

screen.blit(five, [50, 400])

screen.blit(PlayerName\_w, [250, 148])

screen.blit(Deaths\_w, [530, 148])

for Loop in range(len(leader\_board)):

screen.blit(font2.render(leader\_board[Loop][1], True, BLUE), [250, 200+50\*Loop])

if Loop == 4:

break

for Loop in range(len(leader\_board)):

screen.blit(font2.render(str(leader\_board[Loop][0]), True, BLUE), [530, 200+50\*Loop])

if Loop == 4:

break

Back = font2.render("Back", True, PURPLE)

screen.blit(Back, [720, 550])

if event.type == pygame.MOUSEBUTTONUP and x > 710 and x < 800 and y > 540 and y < 600:

Page = 0

Leader\_board = []

#-------------------------------- PAGE 4 ------------------------------------------------------

if Page == 4:

Resume = font.render("Resume ", True, GREEN)

Restart = font.render("Restart", True, RED)

Menu = font.render("Menu", True, BLUE)

screen.blit(Resume, [100, 100])

screen.blit(Restart, [100, 200])

screen.blit(Menu, [100, 300])

pygame.draw.rect(screen, BLACK, [450,200,60,200])

pygame.draw.rect(screen, BLACK, [550,200,60,200])

if event.type == pygame.MOUSEBUTTONUP and x > 100 and x < 300 and y > 100 and y < 150:

Page = 1

if event.type == pygame.MOUSEBUTTONUP and x > 100 and x < 300 and y > 200 and y < 250:

x\_coord = 10

y\_coord = 280

Deaths = 0

Page = 1

Level = 1

if event.type == pygame.MOUSEBUTTONUP and x > 100 and x < 300 and y > 300 and y < 350:

x\_coord = 10

y\_coord = 10

Deaths = 0

Level = 1

Page = 0

#------------------------------------ PAGE 5 ------------------------------------------

if Page == 5:

Name = font.render("What is your name?", True, BLACK)

Deaths2 = font.render("Deaths:", True, BLACK)

Deaths3 = font.render(str(Deaths), True, BLACK)

Write = font2.render("Press the letters to write.",True,BLACK)

Write2 = font2.render("Press the right arrow to move to the next square / letter.",True, BLACK)

Write3 = font2.render("Press backspace to delete all the letters.", True, BLACK)

Submit = font.render("Submit",True, BLUE)

screen.blit(Name, [50, 50])

screen.blit(Deaths2, [450, 500])

screen.blit(Deaths3, [650, 500])

screen.blit(Write, [50, 330])

screen.blit(Write2, [50, 380])

screen.blit(Write3, [50, 430])

screen.blit(Submit, [50, 500])

pygame.draw.rect(screen, BLACK,[50,140,700,150],5)

pygame.draw.line(screen, BLACK,[137,140],[137,290],5)

pygame.draw.line(screen, BLACK,[225,140],[225,290],5)

pygame.draw.line(screen, BLACK,[312,140],[312,290],5)

pygame.draw.line(screen, BLACK,[400,140],[400,290],5)

pygame.draw.line(screen, BLACK,[487,140],[487,290],5)

pygame.draw.line(screen, BLACK,[575,140],[575,290],5)

pygame.draw.line(screen, BLACK,[662,140],[662,290],5)

if event.type == pygame.MOUSEBUTTONUP and x > 50 and x < 230 and y > 500 and y < 550:

if os.path.isfile('LeaderBoard.json'):

file = open('LeaderBoard.json', 'r')

leader\_board = json.loads(file.read())

file.close()

else:

file = open('LeaderBoard.json', 'w+')

file.write("[]")

file.close()

pygame.draw.rect(screen, WHITE,[50,500,180,50])

PlayerName = str(No1+No2+No3+No4+No5+No6+No7+No8)

Array = [Deaths, PlayerName]

leader\_board.append(Array)

leader\_board = sorted(leader\_board, key=lambda player: player[0])

with open('LeaderBoard.json', 'w') as f:

f.write(json.dumps(leader\_board))

Page = 3

if event.type == pygame.MOUSEBUTTONUP and x > 450 and x < 800 and y > 500 and y < 550:

Page = 3

if event.type == pygame.KEYUP:

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

find\_letter("A")

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

find\_letter("B")

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

find\_letter("C")

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

find\_letter("D")

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

find\_letter("E")

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

find\_letter("F")

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

find\_letter("G")

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

find\_letter("H")

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

find\_letter("I")

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

find\_letter("J")

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

find\_letter("K")

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

find\_letter("L")

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

find\_letter("M")

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

find\_letter("N")

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

find\_letter("O")

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

find\_letter("P")

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

find\_letter("Q")

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

find\_letter("R")

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

find\_letter("S")

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

find\_letter("T")

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

find\_letter("U")

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

find\_letter("V")

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

find\_letter("W")

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

find\_letter("X")

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

find\_letter("Y")

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

find\_letter("Z")

if event.key == pygame.K\_RIGHT and Letter == 1 and No1 != "":

Letter = 2

if event.key == pygame.K\_RIGHT and Letter == 2 and No2 != "":

Letter = 3

if event.key == pygame.K\_RIGHT and Letter == 3 and No3 != "":

Letter = 4

if event.key == pygame.K\_RIGHT and Letter == 4 and No4 != "":

Letter = 5

if event.key == pygame.K\_RIGHT and Letter == 5 and No5 != "":

Letter = 6

if event.key == pygame.K\_RIGHT and Letter == 6 and No6 != "":

Letter = 7

if event.key == pygame.K\_RIGHT and Letter == 7 and No7 != "":

Letter = 8

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

Letter = 1

No1 = ""

No2 = ""

No3 = ""

No4 = ""

No5 = ""

No6 = ""

No7 = ""

No8 = ""

if Letter == 1:

pygame.draw.line(screen,BLACK,[60,270],[127,270],10)

if Letter == 2:

pygame.draw.line(screen,BLACK,[147,270],[215,270],10)

if Letter == 3:

pygame.draw.line(screen,BLACK,[235,270],[302,270],10)

if Letter == 4:

pygame.draw.line(screen,BLACK,[322,270],[390,270],10)

if Letter == 5:

pygame.draw.line(screen,BLACK,[410,270],[477,270],10)

if Letter == 6:

pygame.draw.line(screen,BLACK,[497,270],[565,270],10)

if Letter == 7:

pygame.draw.line(screen,BLACK,[585,270],[652,270],10)

if Letter == 8:

pygame.draw.line(screen,BLACK,[672,270],[739,270],10)

if No1 != "":

screen.blit(L\_font.render(No1, True, BLACK), [73, 185])

if No2 != "":

screen.blit(L\_font.render(No2, True, BLACK), [160, 185])

if No3 != "":

screen.blit(L\_font.render(No3, True, BLACK), [247, 185])

if No4 != "":

screen.blit(L\_font.render(No4, True, BLACK), [335, 185])

if No5 != "":

screen.blit(L\_font.render(No5, True, BLACK), [422, 185])

if No6 != "":

screen.blit(L\_font.render(No6, True, BLACK), [510, 185])

if No7 != "":

screen.blit(L\_font.render(No7, True, BLACK), [597, 185])

if No8 != "":

screen.blit(L\_font.render(No8, True, BLACK), [685, 185])

#---------------------------- PAGE 6 ----------------------------------------------

if Page == 6:

if Level == 2:

screen.blit(L\_font.render("Dont even bother trying", True, BLACK), [50, 250])

Count += 1

if Count == 400:

Page = 1

Count = 0

if Level == 3:

screen.blit(L\_font.render("Harder than it looks", True, BLACK), [50, 250])

Count += 1

if Count == 400:

Page = 1

Count = 0

if Level == 4:

screen.blit(L\_font.render("Hell, why not", True, BLACK), [50, 250])

Count += 1

if Count == 400:

Page = 1

Count = 0

if Level == 5:

screen.blit(L\_font.render("Now it gets hard", True, BLACK), [50, 250])

Count += 1

if Count == 400:

Page = 1

Count = 0

# --- Go ahead and update the screen with whats been drawn

pygame.display.flip()

# --- Limit to 60 frames per second

clock.tick(180)

#Close the window and quit

pygame.quit()