from itertools import cycle

from random import randrange

from tkinter import Canvas, Tk, messagebox, font

canvas\_width = 800

canvas\_height = 400

root = Tk()

root.title("Egg Catcher")

c = Canvas(root, width=canvas\_width, height=canvas\_height, background="deep sky blue")

c.create\_rectangle(-5, canvas\_height-100, canvas\_width+5, canvas\_height+5, fill="sea green", width=0)

c.create\_oval(-80, -80, 120, 120, fill='orange', width=0)

c.pack()

color\_cycle = cycle(["light blue", "light green", "light pink", "light yellow", "light cyan"])

egg\_width = 45

egg\_height = 55

egg\_score = 10

egg\_speed = 500

egg\_interval = 4000

difficulty = 0.95

catcher\_color = "blue"

catcher\_width = 100

catcher\_height = 100

catcher\_startx = canvas\_width / 2 - catcher\_width / 2

catcher\_starty = canvas\_height - catcher\_height - 20

catcher\_startx2 = catcher\_startx + catcher\_width

catcher\_starty2 = catcher\_starty + catcher\_height

catcher = c.create\_arc(catcher\_startx, catcher\_starty, catcher\_startx2, catcher\_starty2, start=200, extent=140, style="arc", outline=catcher\_color, width=3)

game\_font = font.nametofont("TkFixedFont")

game\_font.config(size=18)

score = 0

score\_text = c.create\_text(10, 10, anchor="nw", font=game\_font, fill="darkblue", text="Score: "+ str(score))

lives\_remaining = 3

lives\_text = c.create\_text(canvas\_width-10, 10, anchor="ne", font=game\_font, fill="darkblue", text="Lives: "+ str(lives\_remaining))

eggs = []

def create\_egg():

x = randrange(10, 740)

y = 40

new\_egg = c.create\_oval(x, y, x+egg\_width, y+egg\_height, fill=next(color\_cycle), width=0)

eggs.append(new\_egg)

root.after(egg\_interval, create\_egg)

def move\_eggs():

for egg in eggs:

(eggx, eggy, eggx2, eggy2) = c.coords(egg)

c.move(egg, 0, 10)

if eggy2 > canvas\_height:

egg\_dropped(egg)

root.after(egg\_speed, move\_eggs)

def egg\_dropped(egg):

eggs.remove(egg)

c.delete(egg)

lose\_a\_life()

if lives\_remaining == 0:

messagebox.showinfo("Game Over!", "Final Score: "+ str(score))

root.destroy()

def lose\_a\_life():

global lives\_remaining

lives\_remaining -= 1

c.itemconfigure(lives\_text, text="Lives: "+ str(lives\_remaining))

def check\_catch():

(catcherx, catchery, catcherx2, catchery2) = c.coords(catcher)

for egg in eggs:

(eggx, eggy, eggx2, eggy2) = c.coords(egg)

if catcherx < eggx and eggx2 < catcherx2 and catchery2 - eggy2 < 40:

eggs.remove(egg)

c.delete(egg)

increase\_score(egg\_score)

root.after(100, check\_catch)

def increase\_score(points):

global score, egg\_speed, egg\_interval

score += points

egg\_speed = int(egg\_speed \* difficulty)

egg\_interval = int(egg\_interval \* difficulty)

c.itemconfigure(score\_text, text="Score: "+ str(score))

def move\_left(event):

(x1, y1, x2, y2) = c.coords(catcher)

if x1 > 0:

c.move(catcher, -20, 0)

def move\_right(event):

(x1, y1, x2, y2) = c.coords(catcher)

if x2 < canvas\_width:

c.move(catcher, 20, 0)

c.bind("<Left>", move\_left)

c.bind("<Right>", move\_right)

c.focus\_set()

root.after(1000, create\_egg)

root.after(1000, move\_eggs)

root.after(1000, check\_catch)

root.mainloop()