**DAY 22**

#Building the Pong Arcade Game

from turtle import Screen, Turtle

from paddle import Paddle

from ball import Ball

from scoreboard import Scoreboard

import time

screen = Screen()

screen.bgcolor("black")

screen.setup(width=800, height = 600)

screen.title("Pong Game")

screen.tracer(0)

r\_paddle = Paddle((350, 0))

l\_paddle = Paddle((-350, 0))

ball = Ball()

scoreboard = Scoreboard()

screen.listen()

screen.onkey(r\_paddle.go\_up, "Up")

screen.onkey(r\_paddle.go\_down, "Down")

screen.onkey(l\_paddle.go\_up, "w")

screen.onkey(l\_paddle.go\_down, "s")

game\_is\_on = True

while game\_is\_on:

time.sleep(ball.move\_speed)

screen.update()

ball.move()

if ball.ycor() > 280 or ball.ycor() < -280:

ball.bounce\_y()

if ball.distance(r\_paddle) < 50 and ball.xcor() > 320 or ball.distance(l\_paddle) < 50 and ball.xcor() < -320:

ball.bounce\_x()

if ball.xcor() > 380:

ball.reset\_position()

scoreboard.l\_point()

if ball.xcor() < -380:

ball.reset\_position()

scoreboard.r\_point()

screen.exitonclick()

**DAY 29**

#password

from tkinter import \*

from tkinter import messagebox

#password generator

from random import choice, randint, shuffle

def generate\_password():

letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',

'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',

'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

symbols = ['!', '@', '#', '$', '%', '&', '\*', '+', '\_']

password\_letters= [choice(letters) for \_ in range(randint(8, 10))]

password\_symbols = [choice(symbols) for \_ in range(randint(2, 4))]

password\_numbers = [choice(numbers) for \_ in range(randint(2, 4))]

password\_list = password\_letters + password\_numbers + password\_symbols

shuffle(password\_list)

password\_list = password\_list[:max(8, randint(8, 15))]

password = " ".join(password\_list)

password\_entry.delete(0, END)

password\_entry.insert(0, password)

def save():

website = website\_entry.get()

email = email\_entry.get()

password = password\_entry.get()

if len(password) < 8 or len(password) > 15:

messagebox.showinfo("Invalid Password", "Password should be between 8 and 15 characters.")

return

# messagebox.showinfo(title="Title", message="Personal info")

messagebox.showinfo("Password Requirements", "The password should be between 8 and 15 characters.")

messagebox.askquestion(message="Do you want to submit this ?")

if len(website) == 0 or len(password) == 0:

messagebox.showinfo(title="Fill the empty boxes")

else:

is\_ok = messagebox.askquestion(title="Ttile", message=f" These are the details entered: \n Email: {email} " f"\n Password: {password} \n Is it ok to save?")

if is\_ok:

with open("data.txt", "a") as data\_file:

data\_file.write(f"{website} | {email} | {password}\n")

website\_entry.delete(0, END)

password\_entry.delete(0, END)

window = Tk()

window.title("Password Manager")

window.config(padx=50, pady=50)

cancvas = Canvas(height=200, width=200)

logo\_img = PhotoImage(file="password.png")

cancvas.create\_image(100, 100, image= logo\_img)

cancvas.grid(row=0, column=1)

#labels

website\_label = Label(text="Website:")

website\_label.grid(row=1, column=0)

email\_label = Label(text="Email/Username:")

email\_label.grid(row=2, column=0)

password\_label = Label(text="Password:")

password\_label.grid(row=3, column=0)

#Enteries

website\_entry = Entry(width=50)

website\_entry.grid(row=1, column=1, columnspan=1)

website\_entry.focus()

email\_entry = Entry(width=50)

email\_entry.grid(row=2, column=1, )

email\_entry.focus()

email\_entry.insert(0, "adewalep096@gmail.com")

password\_entry = Entry(width=34,)

password\_entry.grid(row=3, columnspan=2)

#button

generate\_password\_button = Button(text="G/ Passkey", command=generate\_password)

generate\_password\_button.grid(row=3, column=2)

add\_button =Button(text="Add", width=43, command=save)

add\_button.grid(row=4, column=1)

window.mainloop()

**SCOREBOARD.py**

from turtle import Turtle

class Scoreboard(Turtle):

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

super().\_\_init\_\_()

self.color("white")

self.penup()

self.hideturtle()

self.l\_score = 0

self.r\_score = 0

self.update\_scoreboard()

def update\_scoreboard(self):

self.clear()

self.goto(-100, 200 )

self.write(self.l\_score, align = "center", font =("Courier", 70, "normal"))

self.goto(100, 200)

self.write(self.r\_score, align = "center", font =("Courier", 70, "normal"))

def l\_point(self):

self.l\_score += 1

self.update\_scoreboard()

def r\_point(self):

self.r\_score += 1

self.update\_scoreboard()

**PADDLE.py**

from turtle import Turtle

class Paddle(Turtle):

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

super().\_\_init\_\_()

self.shape("square")

self.color("white")

self.shapesize(stretch\_wid=5, stretch\_len=1)

self.penup()

self.goto(position)

def go\_up(self):

new\_y = self.ycor() + 20

self.goto(self.xcor(), new\_y)

def go\_down(self):

new\_y = self.ycor() - 20

self.goto(self.xcor(), new\_y)

**MAIN.py**

#Creating Lists using List Comprehension

#For Loop

numbers = [1, 2, 3]

new\_list = []

for n in numbers:

add\_1 = n + 1

new\_list.append(add\_1)

#List Comprehension

new\_list = [n + 1 for n in numbers]

#String as list

name = "Angela"

letters\_list = [letter for letter in name]

#Range as List

range\_list = [num \* 2 for num in range(1,5)]

#Conditional List Comprehension

names = ["Alex", "Beth", "Caroline", "Dave", "Elanor", "Freddie"]

#List Case Conversion

short\_names = [name for name in names if len(name) < 5]

#List Case Conversion

long\_names = [name.upper() for name in names if len(name) > 5]