

IGNITE COHORT 2
JEMIMAH WANGECI
WEEK 2 ASSIGNMENT

Be creative, Try to use images and good prompts for the AI(chatgpt, claude.ai or copilot) to help you build your own little fun game.

ASSIGNMENT

**Create a game like Ping Pong, or
Guess the Number.**

- If using python for display/GUI use this libraries: [pyqt5 · tkinter · pygui · kivy · wxpython · pysimplegui · libavg · pyforms etc] or use JAVASCRIPT NODE.Js or html and css the one your comfortable with.

NB: Submissions is record a 1 minute Video of the code and you playing the Game. and a document explaining the process you took creating the game and also the prompts used in the game creation.

THE VIDEO SHOULD BE UPLPOADED IN Google Drive and copy the link in the document, make sure the link is shareable so that anyone can be able to see the game if you have a YouTube channel you can also upload and share link, Or even LinkedIn. wherever works for you.

#BE-CREATIVE-BE-SMART-USE-AI

PROCESS OF CREATING THE PING PONG GAME

1. Setting Up the Environment

- **Installed Python:** Ensured Python was installed on the computer.
- **Installed tkinter:** Used the command `pip install tk` to install the tkinter library for GUI development.

2. Creating the Game Window

- **Initialized the Main Window:** Created the main window using `tkinter.Tk()`.
- **Configured the Window:** Set the title, disabled resizing, and ensured the window stays on top.

3. Adding the Paddles and Ball

- **Paddle Class:** Created a Paddle class to handle paddle creation and movement.
 - **Attributes:** Included attributes for position, movement, and score.
 - **Methods:** Added methods to draw the paddle and handle left and right movements.
- **Ball Class:** Created a Ball class to handle ball creation and movement.
 - **Attributes:** Included attributes for position and movement.
 - **Methods:** Added methods to draw the ball, detect collisions with paddles, and update scores.

4. Animating the Game

- **Game Loop:** Created a game loop to continuously update the positions of the ball and paddles.

5. Adding Score Tallying

- **Score Attributes:** Added score attributes to the Paddle class.
- **Score Display:** Used `canvas.create_text` to display the scores on the canvas.

6. Adding Pause and Resume Functionality

- **Pause Variable:** Introduced a paused variable to control the game state.
- **Pause Function:** Created a `pause_game` function to toggle the paused state and update the button text.

7. Adding Restart Functionality

- **Restart Function:** Created a `restart_game` function to reset the game state, recreate the paddles, ball, scoreboard, and buttons.

8. Adding On-Screen Buttons

- **Button Frame:** Created a `button_frame` to hold the buttons and positioned it at the top of the window.

- **Stylish Buttons:** Used `tkk.Button` for a modern look and added padding for better spacing.

Prompts Used in the Game Creation

1. **Initial Setup and Game Window:**
 - “How do I create a simple game window using tkinter in Python?”
 - “What are the basic steps to set up a tkinter window for a game?”
2. **Creating Paddles and Ball:**
 - “How can I create a paddle class in Python using tkinter?”
 - “What methods should I include in a ball class for a Ping Pong game?”
3. **Animating the Game:**
 - “How do I create a game loop in Python to continuously update the game state?”
4. **Adding Score Tallying:**
 - “How can I add score tallying to a Ping Pong game in Python?”
 - “What is the best way to display scores on a tkinter canvas?”
5. **Pause and Resume Functionality:**
 - “How can I implement a pause and resume feature in a Python game?”
 - “What is the best way to toggle the game state using a button in tkinter?”
6. **Restart Functionality:**
 - “How can I reset the game state and restart a game in Python?”
 - “What steps are needed to recreate game objects in a restart function?”
7. **On-Screen Buttons:**
 - “How can I add on-screen buttons to a tkinter game?”
 - “What are some ways to make tkinter buttons look more modern and stylish?”

CODE EXPLAINED IN DETAIL

1. Setting Up the Environment

```
import tkinter as tk
from tkinter import ttk
import time
```

tkinter: This is the standard Python interface to the Tk GUI toolkit.

ttk: This module provides access to the Tk themed widget set, which gives a more modern look to the widgets.

time: This module provides various time-related functions, used here for controlling the game loop.

2. Creating the Paddle Class

```
class Paddle:
    def __init__(self, canvas, color, x, y, left_key, right_key):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0, 0, 100, 10, fill=color)
        self.canvas.move(self.id, x, y)
        self.x = 0
        self.score = 0 # Initialize score attribute
        self.canvas_width = self.canvas.winfo_width()
        self.canvas.bind_all(left_key, self.turn_left)
        self.canvas.bind_all(right_key, self.turn_right)

    def draw(self):
        self.canvas.move(self.id, self.x, 0)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:
            self.x = 0

    def turn_left(self, evt):
        self.x = -2

    def turn_right(self, evt):
        self.x = 2
```

init method: Initializes the paddle with its position, color, and key bindings for movement.

draw method: Moves the paddle based on the current x-direction and ensures it stays within the canvas boundaries.

turn_left and turn_right methods: Change the paddle's x-direction when the corresponding keys are pressed.

3. Creating the Ball Class

```
class Ball:
    def __init__(self, canvas, paddle1, paddle2, color):
        self.canvas = canvas
        self.paddle1 = paddle1
        self.paddle2 = paddle2
        self.id = canvas.create_oval(0, 0, 15, 15, fill=color)
        self.canvas.move(self.id, 400, 300)
        self.x = 1
        self.y = -1
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()
        self.hit_bottom = False

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)
        if pos[1] <= 0:
            self.y = 1
            self.paddle2.score += 1
            self.canvas.itemconfig(score2_text, text=f"Player 2: {self.paddle2.score}")
        if pos[3] >= self.canvas_height:
            self.y = -1
            self.paddle1.score += 1
            self.canvas.itemconfig(score1_text, text=f"Player 1: {self.paddle1.score}")
        if pos[0] <= 0 or pos[2] >= self.canvas_width:
            self.x = -self.x
        self.hit_paddle(pos)

    def hit_paddle(self, pos):
        paddle1_pos = self.canvas.coords(self.paddle1.id)
        paddle2_pos = self.canvas.coords(self.paddle2.id)
        if pos[2] >= paddle1_pos[0] and pos[0] <= paddle1_pos[2]:
            if pos[3] >= paddle1_pos[1] and pos[3] <= paddle1_pos[3]:
                self.y = -self.y
        if pos[2] >= paddle2_pos[0] and pos[0] <= paddle2_pos[2]:
```

```
if pos[1] <= paddle2_pos[3] and pos[1] >= paddle2_pos[1]:
    self.y = -self.y
```

init method: Initializes the ball with its position, color, and references to the paddles.

draw method: Moves the ball, checks for collisions with the canvas boundaries, updates scores, and calls hit_paddle to check for paddle collisions.

hit_paddle method: Checks if the ball has hit either paddle and reverses its y-direction if it has.

4. Pause and Restart Functions

```
def pause_game():
    global paused
    paused = not paused
    if paused:
        pause_button.config(text="Resume")
    else:
        pause_button.config(text="Pause")

def restart_game():
    global ball, paddle1, paddle2
    canvas.delete("all")
    paddle1 = Paddle(canvas, "blue", 350, 550, "<Left>", "<Right>")
    paddle2 = Paddle(canvas, "green", 350, 50, "a", "d")
    ball = Ball(canvas, paddle1, paddle2, "red")
    create_scoreboard()
```

pause_game function: Toggles the paused state and updates the button text accordingly.

restart_game function: Resets the game by deleting all canvas items and recreating the paddles, ball, and scoreboard.

5. Creating the Scoreboard and Buttons

```
def create_scoreboard():
    global score1_text, score2_text
```

```

    score1_text = canvas.create_text(50, 30, text=f"Player 1: {paddle1.score}",
font=("Arial", 16), fill="blue")
    score2_text = canvas.create_text(750, 30, text=f"Player 2: {paddle2.score}",
font=("Arial", 16), fill="green")

def create_buttons():
    global pause_button, restart_button
    pause_button = ttk.Button(button_frame, text="Pause", command=pause_game)
    pause_button.pack(side="left", padx=10)
    restart_button = ttk.Button(button_frame, text="Restart", command=restart_game)
    restart_button.pack(side="left", padx=10)

```

create_scoreboard function: Creates text items on the canvas to display the scores of both players.

create_buttons function: Creates the pause and restart buttons and places them in the button_frame.

6. Main Function and Game Loop

```

def main():
    global window, canvas, ball, paddle1, paddle2, paused, button_frame
    paused = False

    window = tk.Tk()
    window.title("Ping Pong Game")
    window.resizable(0, 0)
    window.wm_attributes("-topmost", 1)

    button_frame = tk.Frame(window)
    button_frame.pack(side="top", pady=10)

    canvas = tk.Canvas(window, width=800, height=600, bd=0, highlightthickness=0)
    canvas.pack()
    window.update()

    paddle1 = Paddle(canvas, "blue", 350, 550, "<Left>", "<Right>")
    paddle2 = Paddle(canvas, "green", 350, 50, "a", "d")
    ball = Ball(canvas, paddle1, paddle2, "red")

    create_scoreboard()
    create_buttons()

    while True:

```

```
    if not paused:
        ball.draw()
        paddle1.draw()
        paddle2.draw()
    window.update_idletasks()
    window.update()
    time.sleep(0.01)

if __name__ == "__main__":
    main()
```

main function: Initializes the game window, canvas, paddles, ball, scoreboard, and buttons. It also contains the game loop that continuously updates the game state.

Game Loop: Runs indefinitely, updating the positions of the ball and paddles if the game is not paused.

LINK TO GAME:

https://drive.google.com/file/d/1rY9wtXImXwUjZdheXmxM4Ba5XrPbqQfl/view?usp=drive_link