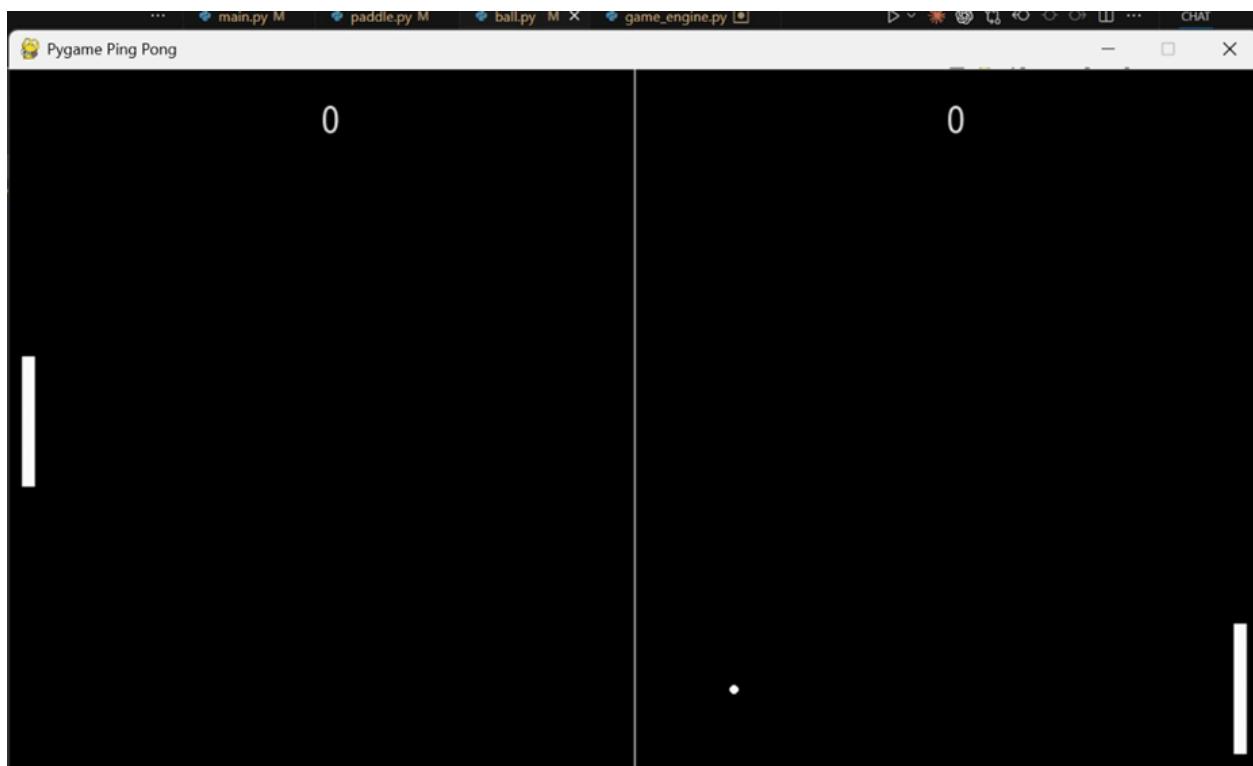


SE Lab 4

Name: Rithvik Rajesh Matta

SRN: PES2UG23CS485

Class: 5H





I'm working on a real-time Ping Pong game using Python and Pygame. I have a partially working project structure. help me with reviewing the logic and also guide me as to how i shd implement the missing features. Review my code and answer subsequent questions

Nice — this is a **clean and modular start** for your real-time Ping Pong game 🎉

Let's break down your current logic, what's working well, and what's missing so you can move forward confidently.

✓ Code Review: Structure & Logic

1. Imports and Initialization

python

Copy code

```
pygame.mixer.pre_init(44100, -16, 2, 512)
pygame.init()
pygame.mixer.init()
```

✓ Good practice — `pre_init()` reduces latency before initializing pygame, especially for sound effects.

Just make sure your sound assets are loaded only after `pygame.mixer.init()` (likely handled in your engine).

Task 1

Please help me with my ping pong game's ball collision. Sometimes the ball goes through paddles. When it occurs, I must reverse `velocity_x` and see if the ball's rectangle overlaps with the paddle rectangles. Add the collision check as soon as you move the ball, and it should function flawlessly at high speeds.

Perfect — that's a **classic collision tunneling issue**, and you're thinking in exactly the right direction 🎉

When your ball moves at high speed, it can "skip" over paddle rectangles between frames — meaning it *never* technically "collides" because the positions don't overlap on any single frame.

Let's fix it properly and robustly ✨

Task 2

When a player scores five points, I need a game to play over the screen. Make a method that displays "Player Wins!" or "AI Wins!" on the screen after determining whether either score equals 5. To ensure that players see the message, make sure the game loop is maintained. Before ending the pygame, add a brief delay.

Perfect — that's a clean feature to add: a "Game Over" message when either player reaches 5 points, with a short display delay before exiting or resetting.

You're right — we'll need to:

- Detect when a player's score hits 5
- Display "Player Wins!" or "AI Wins!" (centered)
- Keep the loop running briefly so the player actually *sees* it
- Then quit or reset the game gracefully

✳️ Here's a Method You Can Add ↴ Your GameEngine Class

Task 3

After the game is over, provide a replay feature. "Best of 3", "Best of 5", "Best of 7", or "Exit" alternatives should be displayed. Await input from the user using keys 3, 5, 7, or ESC. Reset the ball location and update the winning score target when they make a decision. They should be able to play again after that.

Perfect 🎉 — you're designing this exactly how real games handle replay menus.

We'll now create a **replay menu system** that:

- Displays after someone wins (Player or AI)
- Offers "Best of 3", "Best of 5", "Best of 7", or "Exit" options
- Waits for user input (3 , 5 , 7 , or ESC)
- Resets the ball, paddles, and scores
- Updates the **target winning score** accordingly
- Lets them immediately play again without restarting the program

Task 4

Enhance my ping pong game in Python with sound effects.
Use pygame.mixer to load.wav files for scoring, wall
bounce, and paddle hit.Sound(). Every time ball.velocity_x
or ball.velocity_y changes, play the sounds. At the
beginning of the file, initialize pygame.mixer.

Excellent — adding **sound effects** will really polish your Ping Pong game 🎵

You already initialized the mixer correctly at the top of your `main.py`, so now we'll integrate the **sound system** into your `GameEngine` (and specifically the ball's movement logic).

Let's walk through a clean and extensible setup ↪

🎧 1. Initialize `pygame.mixer` (already done in your main)

You've already got this:



🎧 1. Initialize `pygame.mixer` (already done in your main)

You've already got this:

```
python
```

Copy code

```
pygame.mixer.pre_init(44100, -16, 2, 512)  
pygame.init()  
pygame.mixer.init()
```

Perfect — low-latency setup for `.wav` sounds.

📁 2. Place your sound files

Make sure your project folder looks like this:

```
css
```

Copy code

```
ping-pong/  
|  
└── main.py  
└── game/
```



Chatgpt link -> <https://chatgpt.com/share/68f11df3-2d34-8001-bc77-69dfe79cdd2e>

Github link -> <https://github.com/rtk5/Ping-pong>