Multiplayer Ping Pong Game with Dynamic Obstacles

Take Home Assessment

Time Limit: 3 Hours

You are allowed and encouraged to use ChatGPT or similar AI tools to help with implementation details and documentation.

Project Overview

Create a real-time multiplayer ping pong game with dynamic obstacles where players can compete across different browser tabs.

Requirements

1. Game Mechanics

- Implement a two-player ping pong game with paddles and a moving ball
- Players should control paddles using keyboard input (choice of keys is up to you)
- Include scoring system when ball goes out of bounds
- Add two square obstacles within the playing field that:
  - Are randomly positioned at game start
  - Don't overlap with the ball's starting position
  - Cause the ball to bounce on collision

2. Backend (Python)

- Create a Python backend server
- Handle game state management
- Implement real-time communication with frontend
- Track obstacle positions and collisions

3. Frontend (JavaScript)

- Use any JavaScript framework/library of your choice
- Create game interface showing:
  - Both players' paddles
  - Moving ball
  - Score

    - Two obstacles
- Allow gameplay in separate browser tabs for each player

4. Real-time Communication

- Implement WebSocket connection between frontend and backend
- Ensure player actions are reflected in real-time for both players
- Handle basic error cases (disconnection, reconnection)

Submission Requirements

1. Code Repository

- Upload to GitHub/GitLab
- Include README.md with:
    - Setup instructions
    - How to run the game
    - Brief explanation of your technical choices
    - Any known limitations given the time constraint

2. Video Walkthrough

- Create a 5-minute (max) screen recording that shows:
    - How to set up and run the application
    - Demonstration of the game being played between two browser tabs
    - Brief explanation of key technical decisions
    - Showcase of the obstacle mechanics
- Video can be uploaded to any accessible platform (YouTube unlisted, Google Drive, etc.)

3. Notes

- Focus on core functionality over visual polish
- Basic styling is sufficient
- Document any major assumptions made
- It's okay to have minor bugs given the time constraint
- Include a brief note about which parts were developed with AI assistance

What Not to Focus On

- Advanced graphics or animations
- Perfect physics
- Complex UI/UX
- Edge cases
- Extensive error handling
- Perfect code organization

Remember: The goal is to demonstrate your ability to create a working full-stack application with real-time features in a limited time frame. Use your time wisely and leverage AI tools to help with implementation details.