**📝 Project Log: Console Pong Game in C#**

**Project Title**

**Pong Console Game with Difficulty Levels**

**Project Goals**

The goal of this project is to design and implement a **text-based Pong game** using the C# console.  
The program demonstrates understanding of:

* Fundamental C# syntax and control structures
* Console input/output handling
* Methods and modular code organization
* Loops, conditionals, and variables
* Real-time game logic and simple rendering
* User interaction and dynamic difficulty control

By the end of the project, the game should provide a smooth, flicker-free gameplay experience with adjustable difficulty settings.

**Development Steps Completed So Far**

**1. Concept & Planning**

* Selected **Pong** as a suitable beginner project combining logic, rendering, and user input.
* Decided to implement it as a **single-player console game** with keyboard controls (W/S or arrow keys).

**2. Basic Game Setup**

* Created a PongGame class to contain all game logic.
* Defined game field dimensions, paddle and ball positions, and speed control variables.
* Implemented initialization (Initialize()) to draw static borders and set starting positions.

**3. Core Game Mechanics**

* Added a **main game loop (Run())** controlling the sequence:  
  Input → Logic → Draw → Sleep.
* Implemented **ball movement and bouncing** off walls.
* Added **paddle control** using keyboard input.
* Created **collision detection** for paddle hits and scoring.

**4. Difficulty Selection**

* Implemented a **menu (ChooseDifficulty())** allowing the player to select between  
  *Easy*, *Medium*, and *Hard* speeds (150ms, 100ms, 60ms per frame).

**5. Flicker-Free Rendering**

* Replaced Console.Clear() calls with **partial screen updates** using Console.SetCursorPosition().
* Implemented logic to **erase old positions** of the ball and paddle before redrawing them,  
  eliminating the flicker problem common in console animation.

**6. Testing and Debugging**

* Verified smooth rendering and consistent paddle/ball behavior.
* Fixed issues with object duplication and paddle “stretching” by adjusting update order.
* Confirmed that pressing ESC exits cleanly and displays the final score.

**Current Features**

* Real-time paddle and ball movement
* Scoring system
* Three difficulty levels
* Smooth, flicker-free console animation
* Clean game-over message

**Next Planned Steps (Future Improvements)**

* Add an **AI opponent** for single-player mode.
* Implement **sound effects** (via Console.Beep()).
* Include **start/restart menu** and score persistence.