Pong Multiplayer Game

**Group Members**

1. Shamveel Khan (24K-0962)  
2. Muzamil Suleman (24K-1023)  
3. Kabeer Javed (24K-0700)

# 1. Executive Summary

**• Overview:**

This project is a multiplayer Pong game developed using Raylib and ENet in C++. The project aimed to implement real-time gameplay mechanics, including syncing ball and paddle positions over a network. It focuses on OOP principles like encapsulation and modularity.

**• Key Findings:**

Achieved smooth synchronization using a self-made aspect ratio handling method, implemented themes, GUI, and networking logic.

# 2. Introduction

**• Background:**

This Pong project serves as a practical implementation of object-oriented programming concepts applied to a classic game with multiplayer capabilities.

**• Project Objectives:**

To develop a feature-rich multiplayer Pong game using Raylib for graphics and ENet for networking.

# 3. Project Description

**• Scope:**

The game includes online multiplayer, theme switching, GUI menu, ball movement syncing, and basic game physics. Advanced AI or matchmaking systems are not included.

**• Technical Overview:**

Developed using Visual Studio Code with GCC. Technologies used: Raylib (graphics), ENet (networking). Assets are organized per theme in respective folders.

# 4. Methodology

**• Approach:**

We used a planned approach inspired by DeepSeek. The first 14 days were dedicated to learning ENet and the next 10 for implementing networking. Development was modular with clear weekly milestones.

**• Roles and Responsibilities:**

Shamveel: Networking, game logic, ball sync logic.  
Muzamil: GUI and menu implementation.  
Kabeer: Themes and assets management.

# 5. Project Implementation

**• Design and Structure:**

The project is designed in a modular way with responsibilities clearly divided among team members.

**• Functionalities Developed:**

Multiplayer support, ball movement synchronization, paddle syncing, GUI with theme selection.

**• Challenges Faced:**

ENet had minimal community support, making it difficult to set up and learn. Learning interpolation and syncing across different resolutions was also tough. Came up with a method to send divided values for syncing, avoiding the need to exchange screen sizes.

# 6. Results

**• Project Outcomes:**

A fully functional multiplayer Pong game with GUI and theme support that effectively uses OOP principles.

**• Screenshots and Illustrations:**

Screenshots are included in the supplementary submission.

**• Testing and Validation:**

Tested across different resolutions and connections to ensure reliable synchronization and responsive gameplay.

# 7. Conclusion

**• Summary of Findings:**

Successfully developed a multiplayer game with real-time synchronization. Learned valuable lessons in networking, interpolation, and game design using Raylib and ENet.

**• Final Remarks:**

This project greatly enhanced our understanding of both OOP and game networking.