

PING PONG GAME.

OOP PROJECT USING C++ AND RAYLIB LIBRARY.



National University of Computing and Emerging Sciences

Date: 5/11/2025
C++ Object Oriented Programming

Maryam 24K-3011
Mehak Kumari 24K-0622
Areesha 24K-0762

Executive Summary

Overview:

The aim of this project was to design and implement a Ping Pong game using Object-Oriented Programming concepts in C++. The game involves two paddles (player and CPU) and a bouncing ball, all developed using the Raylib graphics library.

Key Findings:

The game successfully demonstrates core OOP principles including encapsulation, inheritance, and polymorphism. We implemented real-time graphics, paddle control, collision detection, and basic AI behavior for the CPU paddle.

Introduction

Background:

Object-Oriented Programming (OOP) is a widely used programming paradigm that encourages modular, reusable, and maintainable code. Developing a Ping Pong game allowed us to apply key OOP concepts in a practical and engaging way.

Project Objectives:

- To apply OOP concepts such as classes, inheritance, encapsulation, and polymorphism.
- To develop an interactive and playable game.
- To enhance programming logic, debugging, and problem-solving skills.

Project Description

Scope:

The project includes a game window, player and CPU paddles, a bouncing ball, basic AI for the CPU, and collision detection. Sound effects, advanced scoring, and difficulty levels were excluded due to time constraints.

Technical Overview:

We used C++ as the core programming language and the Raylib library for graphics rendering. Development was done in Visual Studio Code with C Make as the build system.

METHODOLOGY

The group will follow a basic development process, starting with creating a blank screen and game loop. Then, the paddles and ball will be added, along with ball movement and edge collision detection. The player's paddle will be controlled via keyboard, and a simple AI will move the CPU paddle. Paddle ball collision detection and a basic scoring system will be added at the end. Each member will handle different parts to complete the project efficiently.

- **Mehak:** Create blank screen, game loop, draw paddles and ball.
- **Areesha:** Move ball, detect collision with screen edges, Control player paddle
- **Maryam:** CPU paddle AI, paddle-ball collision detection, scoring and sounds.

PROJECT IMPLEMENTATION

Design and Structure:

Ball class handles ball movement.

Paddle class handles user paddle logic.

CPU Paddle inherits from Paddle and adds basic AI.

Main loop updates all objects and handles collisions.

Functionalities Developed:

Ball movement and bounce logic

Player paddle movement using keyboard input

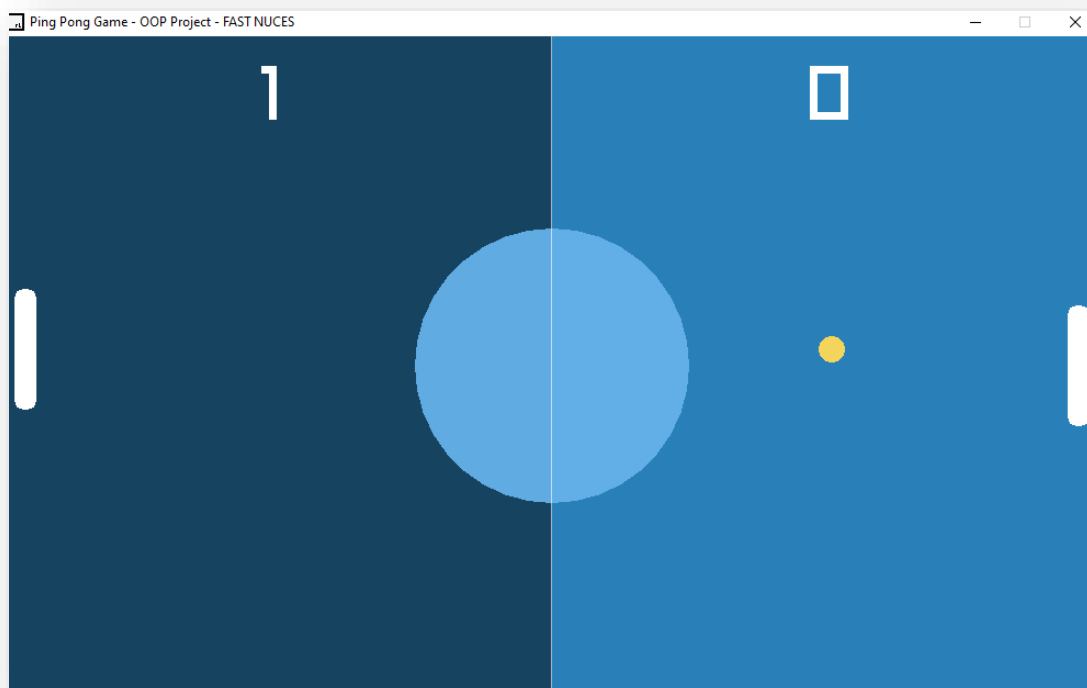
CPU paddle AI movement

Paddle-ball collision detection

Screen boundary checks

Challenges Faced:

Initially faced a compilation error with `cpu.Update(ball.getY())` which was resolved by correctly overloading the `Update` function in the CPU Paddle class. Ensuring smooth paddle movement .



RESULTS

Project Outcomes:

We successfully built a fully functional 2D Ping Pong game using Object-Oriented Programming principles. The game features smooth gameplay, modular code, and clear class separation for the ball, paddles, scoring, and AI. Raylib was used for graphics and input, with raymath.h aiding in collision detection and physics. The result is a clean, maintainable, and engaging game.

Testing and Validation:

We performed extensive manual testing to ensure smooth gameplay, accurate collision detection, and reliable scoring. Each game component (ball, player paddle, CPU AI) was tested individually and in interaction to validate proper behavior. Edge cases—like ball-wall and ball-paddle collisions—were verified under various speeds and positions. We also ensured consistent frame updates and sound playback for a responsive and bug-free user experience.

- Verifying paddle responsiveness with UP/DOWN keys
 - Checking ball behavior against boundaries and paddles
 - Observing CPU paddle tracking accuracy
 - Ensuring no runtime errors occurred
-

We believe the project meets its objectives and serves as a solid foundation for future game development

This project helped us understand how real-time games can be implemented using OOP. We applied inheritance and encapsulation meaningfully and gained experience in graphical programming with Raylib.

