# Project Report: Development of Pong Game.

## Project Title: Pong Game Development: A Classic Reimagined.

---


**Submitted By:**

**Sayantan Saha Utsa** **ID: 2321411642**

**Rayian Rahman Dinan** **ID: 2322915042**

**Sharfin Zaman Swaccha** **ID: 2323713042**


**Submission Date: 02-12-2024**

---

# 1. Introduction.

The Pong Game is widely regarded as one of the first arcade video games and is a symbol of the beginning of the video gaming industry. Developed in the early 1970s, Pong remains a symbol of simplicity and gameplay excellence. In this project, an attempt has been made to recreate this timeless creation using the Java programming language and the Swing library for graphical user interface development. It does provide the core gameplay but also points out important aspects such as user experience, performance optimization, and following modern software development practices.

# 2. Objectives

The main goals of this project are identified as:

- Design and implement a two-player Pong Game with Java and Swing.
- Ensuring that the paddles and the ball move smoothly along with proper collision detection.
- To design an intuitive UI that allows users to start, restart, or quit the game.
- To implement a real-time score tracking system.
- To explore and apply object-oriented programming principles to ensure modularity and maintainability.
- To provide a seamless UX with minimum bugs and maximum responsiveness.

# 3. Project Scope

The scope of this project is the development of the basic version of the Pong Game. It involves:

- A Main Menu interface for starting or exiting the game.
- Two-player gameplay, with separate controls for each player's paddles.
- Ball physics: enabling collision detection with paddles, walls, and scoring areas.
- Real-time scoreboard: tracking the players' scores at all times in the game.

Future development could also include AI gameplay, different themes, and online multiplayer.

# 4. System Requirements

## Software Requirements:

- The game is developed in Java, using the Swing library for GUI. The minimum required version is JDK 8 or higher.
- For development, any IDE supporting Java will work (e.g., IntelliJ IDEA, Eclipse, or NetBeans).
- The game can be run on Windows, macOS, or Linux.

## Hardware Requirements:

- Processor: At least 1.5 GHz dual-core
- RAM: 2 GB
- Storage: 500 MB of free space
- Display: Resolution of 1280x720 or higher

# 5. Design and Implementation

## System Architecture

The Pong Game utilizes a modular object-oriented design in which each part of the game has its own class. These include:

- **MainMenu**: Handles the first menu screen to start or quit the game.
- **GameFrame**: Instantiates the game window and builds the main game area.
- **GamePanel**: Handles game operations like graphics, ball and paddle movement, collision detection, and scoring.
- **Paddle**: Represents the paddles controlled by players and handles user input.
- **Ball**: Controls the movement of the ball and its interaction with the paddles and walls.
- **Score**: Maintains and displays the scores of both players.

- **Class Implementation**

Each class plays a crucial role:

- **Paddle class**: Defines how paddles move depending on the player's input.
- **Ball class**: Creates the movement of the ball and manages its collisions.
- **Score class**: Maintains and shows the scores.
- **GamePanel class**: Ensures smooth game flow by integrating all the elements together.

# 6. User Interface (UI) Design

## Main Menu

- The main menu is simple and intuitive. It displays the game's title at the top of the screen and two buttons below: Start (to begin the game) and Quit (to exit).

## During Gameplay

- **Background**: Black with a white dividing line down the middle.
- **Paddles**: Blue for Player 1 and Red for Player 2. They respond to keyboard input and move vertically.
- **Ball**: A white circle that bounces off walls and paddles, traveling at varying speeds.
- **Scoreboard**: Displays the current scores of both players at the top of the screen.

# 7. Testing and Results

## Testing Strategy

The game underwent rigorous testing:

- **Unit Testing**: Each class was individually tested.
- **Integration Testing**: Ensured all elements functioned together correctly.
- **User Testing**: A small group of players tested the game and provided feedback.

### Test Results

The game performed as expected:

- Paddles moved correctly, the ball path was accurate, and scoring was tracked properly.
- No major bugs were found, and the game loop ran smoothly, even during fast action.

# 8. Challenges and Solutions

### Challenge 1: Collision Detection

- The ball didn't behave correctly when reaching the edges of the paddles. This led to inconsistent gameplay.
- **Solution**: Improved collision detection, adjusting ball speed and direction upon hitting the paddles or walls, and added boundary checks to prevent paddles from moving off-screen.

### Challenge 2: User Input Handling

- Paddles were sometimes slow to respond to input.
- **Solution**: Used the `KeyListener` class and refined the game loop to ensure immediate paddle movement upon pressing the respective keys.

### Challenge 3: Performance Optimization

- The game struggled with high-speed ball movement, affecting responsiveness.
- **Solution**: Refined the game loop to cap the frame rate, ensuring smooth drawing of objects.

# 9. Future Enhancements

- **AI Player**: An AI-controlled player for single-player mode, with varying difficulty levels.
- **Customizable Themes**: Allow players to personalize the game's appearance, including paddles, ball, and background.

- **Online Multiplayer**: Implement online multiplayer to allow players to compete over a network.

# 10. Conclusion

This project successfully recreated the classic Pong game using modern programming techniques. It provides smooth gameplay, real-time scoring, and an easy-to-use interface. The game is a solid foundation for further improvements, such as adding AI and online multiplayer. The use of Java and object-oriented principles ensured that the game was modular and easy to maintain, laying the groundwork for future enhancements.